Transportation Research Part C 116 (2020) 102624

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Region-wide congestion prediction and control using deep learning

Sudatta Mohanty[a,*], Alexey Pozdnukhov[b], Michael Cassidy[c]

[a] University of California, 116, McLaughlin Hall, Berkeley, CA 94720, United States
[b] University of California, 115, McLaughlin Hall, Berkeley, CA 94720, United States
[c] University of California, 416, McLaughlin Hall, Berkeley, CA 94720, United States

A B S T R A C T

Traffic congestion is forecast for neighborhoods within a region using a deep learning model. The model is based on Long Short-Term Memory (LSTM) neural network architecture. It forecasts a congestion score, defined as the ratio of the vehicle accumulation inside a neighborhood to its trip completion rate. Inputs include congestion scores measured at earlier times in neighborhoods within a region, and three other real-time measures of regional traffic.

The ideas are tested using Newell's simplified theory of kinematic waves. Simplified street networks are featured first. Initial tests demonstrate the suitability of the congestion score for characterizing neighborhood traffic conditions, and that the score can be predicted using the four inputs. Further tests of the simplified networks illustrate the value of the deep learning approach, as compared against the use of three benchmark models. A next round of tests shows that the model can be made robust, even to adverse settings. A final round of tests features a pared-down version of the freeway network in the San Francisco Bay Area. The final tests show that the model is scalable. The model is thereafter improved by representing the inputs through weighted undirected graphs that incorporate the route-choice of individuals, and learning features through graph convolutions. A framework for better interpreting the contributions of the model's inputs to its output is developed. A demonstration of the model's usefulness in designing traffic control schemes is presented as well.

1. Introduction

Numerous ideas have been proposed over the years for managing traffic congestion over moderately large spatial scales comparable to a neighborhood or small city; e.g. see (Robertson and David Bretherton, 1991; Diakaki et al., 2002; Papageorgiou et al., 2003; Hale, 2005; Aboudolas et al., 2009; Lin et al., 2012). Many of these ideas feature the use of Macroscopic Fundamental Diagrams (MFDs); e.g. (Daganzo, 2007; Daganzo and Geroliminis, 2008; Geroliminis and Daganzo, 2008; Geroliminis and Levinson, 2009; Zheng et al., 2012; Gonzales and Daganzo, 2012; Keyvan-Ekbatani et al., 2012; Haddad and Geroliminis, 2012; Daganzo et al., 2012; Geroliminis et al., 2013; Yildirimoglu and Geroliminis, 2013; Aboudolas and Geroliminis, 2013; Knoop et al., 2013; Hajiahmadi et al., 2013; Gayah et al., 2014; Ramezani et al., 2015; Hajiahmadi et al., 2015; Haddad and Mirkin, 2016; Girault et al., 2016; Keyvan-Ekbatani et al., 2016; Jusoh and Ampountolas, 2017; Ni and Cassidy, 2018). Re-scaled counterparts to these neighborhood-wide models known as Network Exit Functions (NEFs) have been used as well (Daganzo, 2007; Daganzo and Geroliminis, 2008; Ortigosa et al., 2015). A model of this latter type relates a neighborhood's vehicle accumulations to its trip completion rates, like the example in Fig. 1. Both the MFD and NEF can provide simple, but physically-realistic descriptions of congested states when traffic is homogeneously-loaded over the neighborhood, and is nearly in steady-state (Daganzo and Geroliminis, 2008).

Restricting vehicle inflows to cordoned neighborhoods is one of the congestion-management strategies commonly pursued with these models. In this realm, MFDs and NEFs have been used to determine optimal cordon tolls in support of congestion-pricing schemes (Geroliminis and Levinson, 2009; Gonzales and Daganzo, 2012; Zheng et al., 2012; Yildirimoglu and Geroliminis, 2013;

* Corresponding author.
E-mail addresses: sudatta.mohanty@berkeley.edu (S. Mohanty), alexeip@berkeley.edu (A. Pozdnukhov), cassidy@ce.berkeley.edu (M. Cassidy).

https://doi.org/10.1016/j.trc.2020.102624
Received 28 September 2019; Received in revised form 15 February 2020; Accepted 15 March 2020
Available online 08 May 2020
0968-090X/

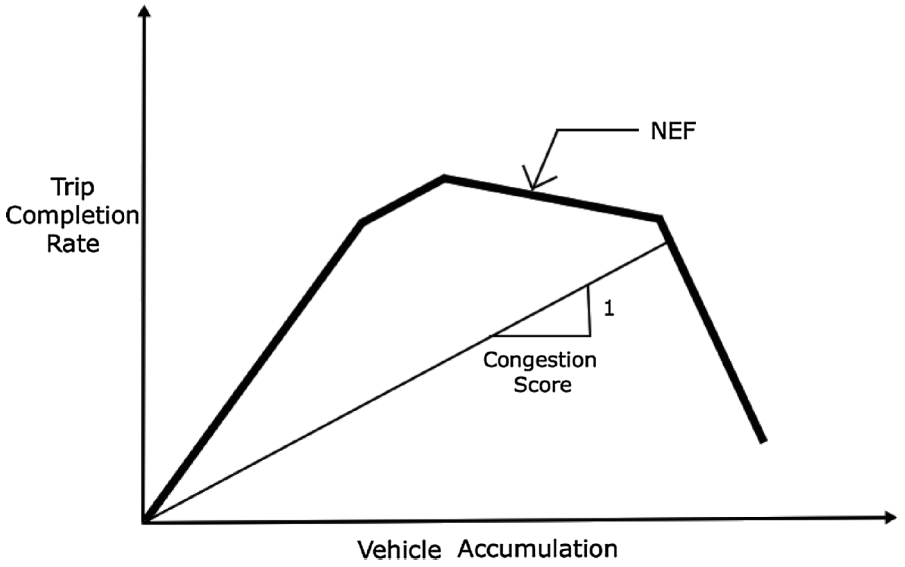


Fig. 1. Example Network Exit Function.

Hajiahmadi et al., 2015). More often, they have guided theorists in re-timing traffic signals to function as cordon meters (Haddad and Geroliminis, 2012; Geroliminis et al., 2013; Aboudolas and Geroliminis, 2013; Knoop et al., 2013; Hajiahmadi et al., 2013; Gayah et al., 2014; Ramezani et al., 2015; Hajiahmadi et al., 2015; Haddad and Mirkin, 2016; Girault et al., 2016; Keyvan-Ekbatani et al., 2016; Jusoh and Ampountolas, 2017; Ni and Cassidy, 2018). In some of these latter cases, the models were combined with flow conservation laws to forecast traffic states in and around cordoned neighborhoods (Haddad and Geroliminis, 2012; Geroliminis et al., 2013; Aboudolas and Geroliminis, 2013; Knoop et al., 2013). The forecasts were used with PID- or model predictive controllers to optimize cordon metering rates. In a handful of these studies (Ramezani et al., 2015; Haddad and Mirkin, 2016; Keyvan-Ekbatani et al., 2016; Jusoh and Ampountolas, 2017), metering rates were varied both over time and over location along a cordon line. In one case (Ni and Cassidy, 2018), this was done using Deep Reinforcement Learning; e.g. (Sutton et al., 1998). Simulations show this deep learning approach to be a promising one.

1.1. Present work

The present research uses a deep learning approach to forecast neighborhood congestion from region-wide observations. The aim is to enhance congestion management on neighborhood-wide scales. The resulting system uses measurable data to estimate regional traffic features that impact neighborhood congestion.

Four traffic features are estimated in all, and are henceforth termed “signals”. They are: (i) the region’s travel demands disaggregated by origin and destination (OD); (ii) average vehicle travel times on the region’s street links; (iii) vehicle accumulations on those links; and (iv) the state of congestion in and around a targeted neighborhood. The fourth signal is expressed as a congestion score. It is defined as the ratio of the neighborhood’s vehicle accumulation to its trip completion rate. The score is thus obtained using the neighborhood’s NEF, as per the example in Fig. 1.

Forecasts of a neighborhood’s future congestion states are expressed as that same score. This is because it is more convenient to predict the scalar (i.e. the ratio of accumulation to trip completion rate) than to predict the two measures separately.[1] Forecasting neighborhood scores from regional signals is a complicated task nonetheless. In addition to the extended spatial scales to be considered, forecasts will need to occur over longer time horizons. The lengthier horizons are required to accommodate the time lags between a signal’s emanation and its eventual impact on the (possibly distant) neighborhood.

In light of these complexities, the task is presently pursued through a deep learning method based on Long Short-Term Memory (LSTM) architecture. In its favor, the LSTM neural network architecture automatically identifies recurrent patterns in data. It does so by means of a gated structure to memorize and store the recurrences over long time periods. This makes LSTM ideal for forecasting the recurrent tendencies in region-wide traffic. The architecture also accommodates deviations in data patterns that are identified from more recent trends. This enables an LSTM model to handle deviations in regional traffic brought by random changes in travel demand (Okutani and Stephanedes, 1984; Chang and Jileng, 1994; Zhou and Mahmassani, 2007), roadway incidents (Ceder, 1982; Golob and Recker, 2003) and the driver route-choice behaviors that these deviations can trigger (Arnott et al., 1993; Arnott and Small, 1994). The LSTM’s gated structure conveniently discards seemingly un-useful information to boot. As compared to traditional machine learning techniques, the LSTM handles very high dimensionality of correlated inputs, without human-engineered feature

[1] The key features affecting the two measures at any given time might be different, which can make predictions more challenging.

extraction (Arel et al., 2010).

The neighborhood congestion score is defined and tested in Section 2. An LSTM model for forecasting neighborhood scores is presented and tested in Section 3. A framework for better interpreting the contribution of model inputs to the output is developed in Section 4. The model is improved further in Section 5 by constructing weighted graphs to represent the inputs and learning features through graph convolutions. A demonstration of the model's usefulness is presented in Section 6. Practical implications of the above efforts are discussed in Section 7.

2. Congestion score

The definition for neighborhood congestion score is presented in Section 2.1. Simulation tests of small, highly-idealized regional networks are presented in Section 2.2. These tests show that the score nicely characterizes a neighborhood's congestion states. The tests also show that the four select signals are suitable proxies for deviant traffic patterns that commonly occur on regional scales, and that can impact neighborhood congestion.

2.1. Neighborhood scoring function

Congestion scores can be assigned to each of Z zones or neighborhoods in a regional network. Assume that the network is partitioned in such ways that traffic is at all times homogeneously loaded across each neighborhood z, with steady state conditions (Ji and Geroliminis, 2012); and that each i^{th} street link lies fully within its (single) neighborhood. Let $A_i(t)$ and $D_i(t)$ be the cumulative numbers of vehicles to have entered and departed link i by time t, respectively, such that $n_i(t) = A_i(t) - D_i(t)$ is the number of vehicles (i.e. the *accumulation*) on link i at time t. Accumulation in neighborhood z, $n^z(t)$, is defined as

$$n^z(t) = \sum_{i \in Z} n_i(t). \tag{1}$$

Let $E_i(t)$ be the cumulative number of trips that ended on link i. If the street links in z are of uniform length, the total trip completion rate in z, $T^z(t)$, is defined as[2]

$$T^z(t) = \sum_{i \in z} \frac{\partial E_i(t)}{\partial t}. \tag{2}$$

Eqs. (1) and (2) are the inputs and outputs of z's NEF respectively, and define the traffic state in z (Daganzo, 2007). We therefore take $\xi^z(t)$ to be the time-dependent congestion score in neighborhood z, defined as

$$\xi^z(t) = \begin{cases} 0, & T^z(t) < \tau^z, \ n^z(t) < \alpha^z \\ n^z(t)/T^z(t), & otherwise, \end{cases} \tag{3}$$

where τ^z and α^z are z's threshold trip completion rate and accumulation, respectively. Values of $T^z(t)$ and $n^z(t)$ that both fall below their thresholds denote that conditions in z are uncongested with low demand. The $\xi^z(t)$ is set to zero in those instances, since they are of little interest and it makes predicting the score simpler.[3] Practically, the values of τ^z and α^z can be set to the mean value observed during non-peak traffic hours.

Though exact measurements of $\xi^z(t)$ would require arrival and departure counts on all links in z, the assumption of homogeneous loading in each partitioned neighborhood means that estimates can be made by sampling counts from a portion of z's links. This makes it possible to track our NEF-based score even when traffic data from a neighborhood are incomplete.

2.2. Sensitivity analysis

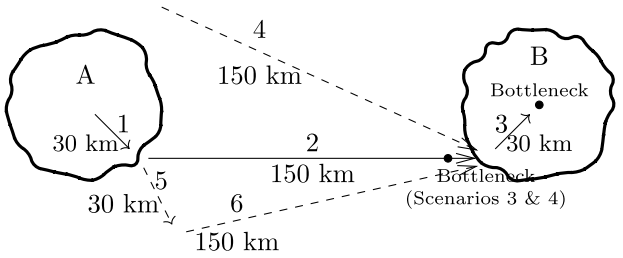
The following simulation experiments demonstrate the congestion score's sensitivity to conditions that emerge on regional networks. They also demonstrate that input signals (i) - (iv) nicely convey how traffic features that emerge throughout a network are carried to neighborhoods to locally impact the scores there. The experiments utilize the simplest, most idealized settings capable of getting the job done; and in all cases roughly emulate morning commutes into a downtown neighborhood.

Experiments start with a baseline network. Its uni-directional links are shown with solid arrows in Fig. 2a. All trips originate from the center of neighborhood A, and are bound for the center of neighborhood B. They are served by the directed links labeled 1 and 3, which reside solely within neighborhoods A and B, respectively; and by the intervening link labeled 2. Notice that each neighborhood's entire street network is represented by a single directed link. The abstraction dispenses with concerns as to whether the neighborhoods are homogeneously-loaded with traffic. This is convenient, since inhomogeneous loadings can render NEFs unusable; see (Daganzo, 2007). The abstraction also means that neighborhood-wide traffic can be described by a single Fundamental Diagram (FD). This makes it easy to model traffic movements using Newell's simplified theory of kinematic waves (Newell, 1993).

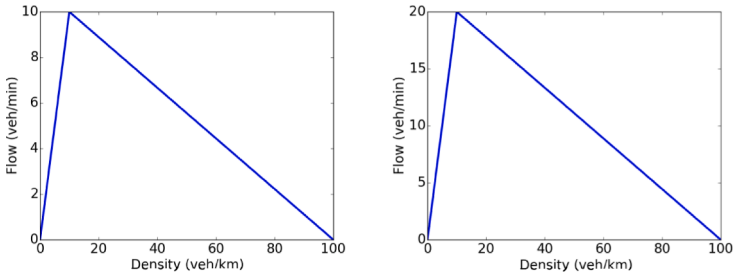
The FDs used for links 1, 2 and 3 are shown in Fig. 2b. These were selected to distinguish relative differences between traffic

[2] Trip completion rate in neighborhoods with unequal link lengths can be calculated as per Edie's generalized definitions (Edie, 1965).

[3] The $\xi^z(t)$ can be smoothed, e.g. by using Locally Weighted Scatterplot Smoothening (LOWESS) (Cleveland, 1981) or nearest neighbor smoothening.



(a) Regional Network (solid links are introduced first in Scenario 1, while dashed links are introduced subsequently)



(b) FD for street links within neighborhoods (left diagram, links 1 and 3) and for arterial links connecting neighborhoods (right diagram, link 2)

Fig. 2. Regional networks and corresponding link fundamental diagrams.

operating in a single lane of a city street (links 1 and 3) from the operations on a single-lane arterial (link 2).[4] A bottleneck with a capacity of 5 vehicles/min is placed at the downstream end of link 3, so as to congest neighborhood B in some cases.

The first scenario will use the baseline settings described above to examine the usefulness of input signal (i), as defined in Section 1. The network will be incrementally altered thereafter to explore the value of the other input signals. Some of those alternations will entail the inclusion of new links. These are shown with dashed arrows in Fig. 2a. Traffic on those links will be modeled using the FD for a single-lane arterial, shown by the right-hand diagram in Fig. 2b.

2.2.1. Scenario 1: time varying ODs, single origin

For the first experiment, travel demand from A to B takes a pulsed shape, one that produces the cumulative demand and the cumulative departure curves in Fig. 3a. The curves form an input-output diagram for link 3. The diagram's vertical displacements are the vehicle accumulations in B (Newell, 1993; Daganzo, 2001).

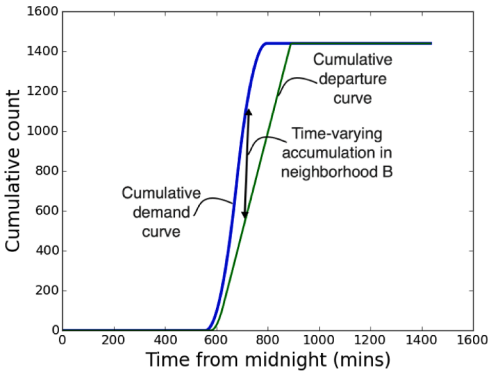
The time-varying pattern of these accumulations is nicely described by B's congestion scores, shown by the solid curve in Fig. 3b. To appreciate this, notice how the scores' sinusoidal shape aligns with the growth and eventual dissipation of the accumulations in neighborhood B. This alignment also underscores how input signal (i), OD pattern, influences B's congestion score. Note from Fig. 3b how the score is shifted in time relative to the demand. The latter is shown by the dotted triangular-shaped curve in that figure. The shift is a function of free flow travel times on links 1–3, which can be inferred by observing the shift between the two curves in Fig. 3b across multiple days.

2.2.2. Scenario 2: time varying ODs, multiple origins

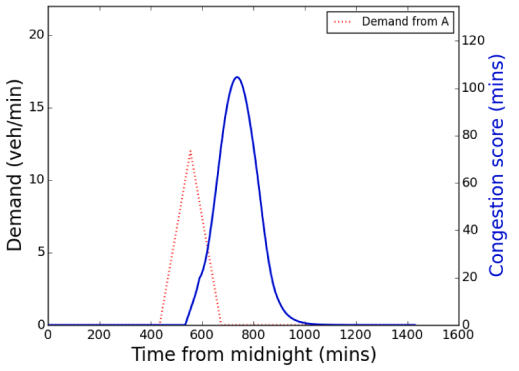
The descriptive power of the congestion score and the influence of signal (i) remains evident when the network becomes more complex. To illustrate, consider a second demand stream that is sent to neighborhood B via link 4; see again Fig. 2a. The second stream displays an identical pulsed pattern, and arrives at B slightly later than the demand from neighborhood A. The two streams coalesce at the entrance to B, and produce the wavy-shaped demand curve in Fig. 4a.

Note neighborhood B's time-varying accumulations, as unveiled in Fig. 4a. Notice how the pattern is nicely aligned with B's congestion scores in Fig. 4b. And note from the latter figure how the shape of that score resembles the pulsed demand curves. One may also note a heightened peak in the score as compared to the scores in the previous scenario. The peak's height is partly governed by the amount of time for which the two demands coalesce at the entrance to neighborhood B. This time is also a function of the two demand patterns, which again underscores the usefulness of input signal (i) for predicting congestion scores.

[4] For the present sensitivity tests, it suffices to use FDs that reflect relative differences in the magnitudes of link capacities, average vehicle speeds, etc.

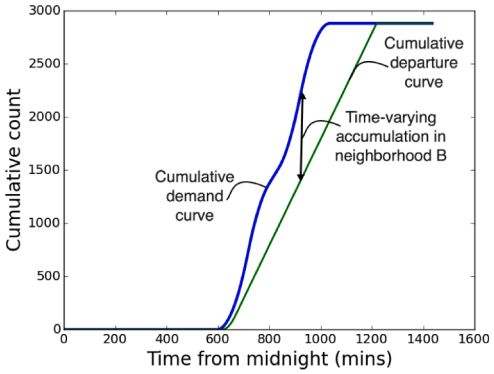


(a) Input-output diagram for link 3 for pulsed demand from a single origin.

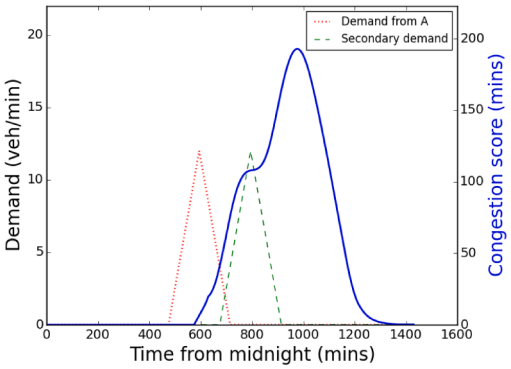


(b) Single pulsed demand from neighborhood A (dotted line, primary y-axis) and the resulting congestion score in neighborhood B (solid line, secondary y-axis) over time.

Fig. 3. The effectiveness of the congestion score and input signal (i) for predicting neighborhood congestion.

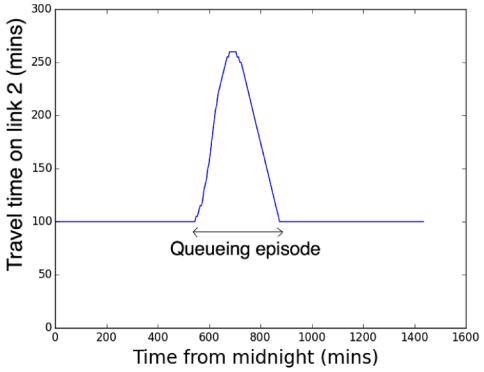


(a) Input-output diagram of link 3 for two pulsed demands that coalesce at the entrance to neighborhood B.

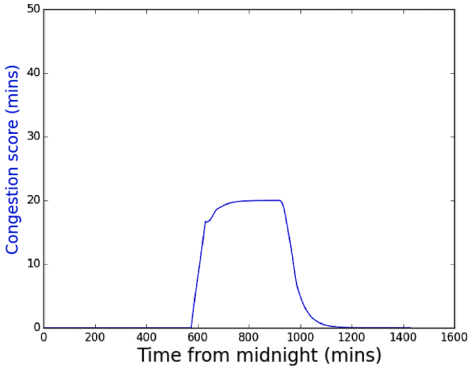


(b) Two pulsed demand streams (dotted and dashed lines, primary y-axis) and congestion score in neighborhood B (solid curve, secondary y-axis) over time.

Fig. 4. Scenario with two pulsed demand streams coalescing at the entrance to neighborhood B.

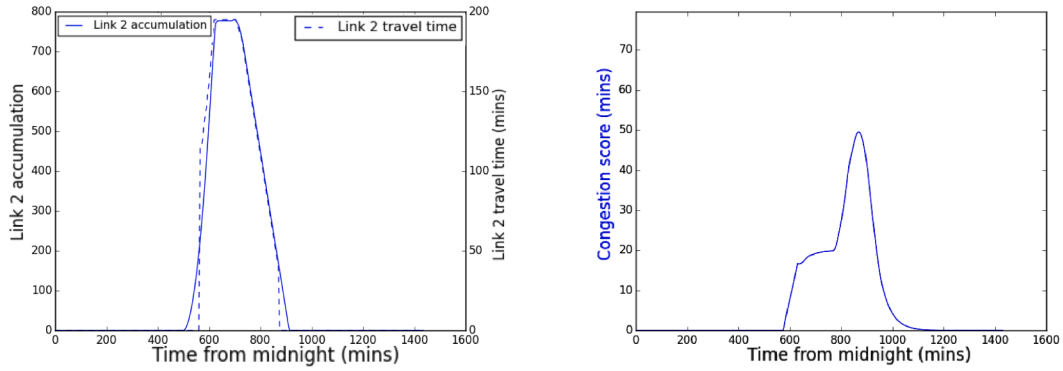


(a) Travel time on link 2 over time



(b) Congestion score vs time

Fig. 5. The usefulness of input signal (ii) for predicting congestion score in the presence of an arterial bottleneck.



(a) Travel times and accumulation on link 2 (b) Congestion score in neighborhood B vs time

Fig. 6. The effectiveness of input signals (ii)-(iii) for predicting congestion score with route switching.

2.2.3. Scenario 3: arterial bottleneck

We return to the baseline network and the original demand pattern, and insert a bottleneck with a capacity of 5 veh/min at the downstream end of link 2; see again Fig. 2a. The new bottleneck generates queueing on link 2, which dissipates after the demand subsides. Telltale signs of this queueing are evident in link 2's travel times, i.e. input signal (ii). A time-series curve for this signal is presented in Fig. 5a. Note how link 2's queueing episode is clearly evident in the figure.

The newly-placed bottleneck causes B's accumulation to level-off during the period when vehicles enter neighborhood B at a rate equal to the capacity of link 2's new bottleneck (5 veh/min). This leveling-off is evident in B's congestion score shown in Fig. 5b. The duration of the level-off period roughly coincides with the duration for which link 2's travel times exceed its free flow travel time.

2.2.4. Scenario 4: arterial bottleneck and route choice

The next experiment features: the baseline network with the newly-added bottleneck on link 2; the original demand pattern; and the addition of links 5 and 6, as shown in Fig. 2a. Since the queue created by the new bottleneck is fully contained on link 2, some commuters will divert to the newly-added links once that bottleneck generates sufficient delay. Commuters are assumed to unilaterally choose routes that minimize their respective travel times; and to hold perfect information to achieve that end (Wardrop and Whitehead, 1952).

The time-series curves for link 2's travel times and accumulations (i.e., input signals (ii) and (iii), respectively) are shown in Fig. 6a. Consideration shows that the plateau displayed by each signal is the period over which diversion takes place via links 5 and 6. The arrivals of diverted vehicles to neighborhood B trigger a surge in the neighborhood's accumulation, one that falls when no more diverted vehicles arrive. Note from Fig. 6b how that surge is nicely captured by B's congestion score. The duration of the plateau in Fig. 6a roughly equals the duration of the surge in Fig. 6b. This underscores the usefulness of signals (ii) and (iii).

2.2.5. Scenario 5: queue spillover

The final test features the baseline settings, but with the lengths of links 2 and 3 shortened to 5 kms each. With these shortened lengths, the queue generated by the bottleneck in Neighborhood B spills over to link 2 and then eventually to link 1. To account for spillovers, the neighborhood analyzed in this test is now A. The congestion scores in B signal the possibility of a queue spillover. The travel times and accumulations on link 2 subsequently confirm that the queue spills over into A.

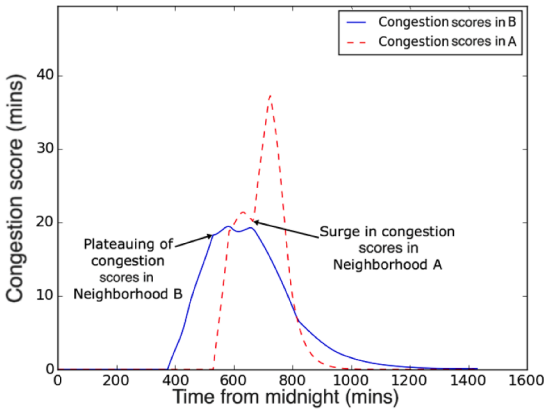
The congestion scores in neighborhood A are shown with the dashed curve in Fig. 7a. The scores are shown in Fig. 7b and c as well. The occurrence of queue spillover in A is nicely captured by the surge in this score. The congestion scores in neighborhood B are shown by the solid curve in Fig. 7a. Note the plateauing of these scores just before the scores surge in A. This plateauing is a telltale sign that a queue has filled neighborhood B and might soon spillover to nearby neighborhoods upstream. A similar plateauing in Fig. 7b and c indicate that the queue has filled Link 2 and may spillover to upstream neighborhood A.

3. Model

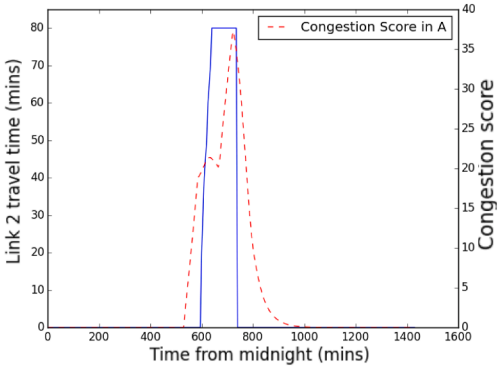
Having shown the value of the four signals, we consolidate them in Section 3.1, so as to reduce the dimensionality (and computing requirements) in the forecasting task. The LSTM model is presented in Section 3.2. The presentation will clarify the sampling intervals to be used in estimating the input signals. The model is put to various tests in Sections 3.3–3.5, as will be described in due course.

3.1. Inputs

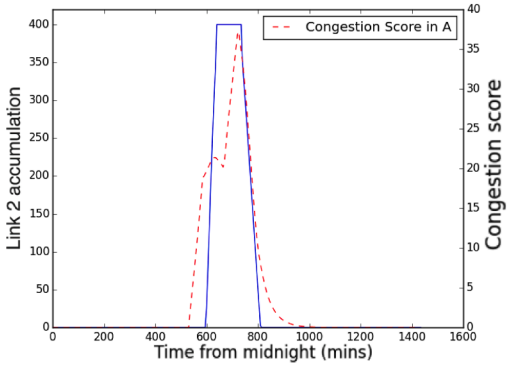
Define a regional network's time-dependent state, $X(t)$, to be the following vector of vectors:



(a) Congestion scores in neighborhood B (solid line) and neighborhood A (dashed line)



(b) Travel times (solid line, primary axis, left) on link 2 along with the congestion scores in neighborhood A (dashed line, secondary axis).



(c) Accumulations (solid line, primary axis, right) on link 2 along with the congestion scores in neighborhood A (dashed line, secondary axis).

Fig. 7. The effectiveness of input signals (ii) - (iv) for predicting congestion scores during queue spillover.

$$X(t) = \begin{bmatrix} D(t) \\ TT(t) \\ C(t) \\ \xi(t) \end{bmatrix}, \tag{4}$$

where the four input vectors are the regional network's:

OD demands between all Z neighborhoods in the network, denoted $D(t)$[5];
vehicle travel times on links from which observations are made, denoted $TT(t)$;
vehicle accumulations on those links, denoted $C(t)$; and congestion scores observed in all Z neighborhoods, denoted $\xi(t)$ and estimated using (3).[6]

We reduce the dimensionality of (4) by limiting $X^z(t)$ to be a vector of elements of $X(t)$ thought to be sufficiently influential in forecasting ξ in neighborhood z. The $D(t)$ were restricted to those with destinations in z[7]; and the elements of $TT(t)$ and $C(t)$ were reduced by filtering-out observations on links that reside at distances more than δ from the centroid of z. The intuition behind this filtering is that the effects of the input signals travel through road links, and that their effect reduces along the path. The value of δ

[5] Real-time OD demand is not directly observable, but can be estimated; see (Zheng et al., 2006; Yang et al., 2017; Mohanty and Pozdnukhov, 2019)

[6] We note for clarity that the observed values of ξ are used as inputs to the model, and forecasted values are outputs.

[7] This reduced the dimensionality of $D(t)$ from $\mathcal{O}(|Z|^2)$ to $\mathcal{O}(|Z|)$

was optimized via cross-validation.

3.2. Deep learning model

The model forecasts congestion scores for neighborhood z, $\hat{\xi}^z$, using the form:

$$[\hat{\xi}^z(t + h^z + p^z), \hat{\xi}^z(t + h^z + p^z - 1), ...,\hat{\xi}^z(t + h^z)] = f(x^z(t), ...,x^z(t - p^z)), \tag{5}$$

where:

f represents a function for an LSTM model;
x^z are values of network state vector, X^z;
h^z is the shortest possible time for an input signal, $x^z(t)$, to travel to the periphery of z; and
p^z is the largest duration over which the signal is estimated to impact congestion within z.

Details regarding the steps followed by the LSTM model are relegated to Appendix A. We note for now that (5) reflects the logic that an input signal observed at t will likely impact z's congestion scores in the interval $[t + h^z, t + h^z + p^z]$. This interval is therefore chosen to be the forecast horizon. The other observed signals which may impact congestion scores in the same time horizon manifest themselves over the period $[t - p^z, t]$. The signals in this period are therefore chosen as the model inputs.

The hyper-parameters h^z and p^z can vary over space and time, but were treated in the present work as constants, so as to limit the number of iterations needed to train the model. The h^z was conservatively chosen to be the signal's minimum free-flow travel time across all possible origins within the network. Similarly, p^z was conservatively taken to be the maximum of all signal durations over space and time. Both estimates were obtained via cross-validation (i.e. the values of h^z and p^z were chosen such that the root mean square prediction error is minimized over a cross-validation dataset).

3.3. Initial tests

Congestion scores forecasted by the deep learning model in (5) are next compared against those from three benchmarks defined below.

(i) 1-Nearest Neighbor(1-NN) model:
The model assumes perfect correlation between congestion scores on consecutive days and is of the form:

$$\xi^z(t + h) = \xi^z(t), \tag{6}$$

where:
$h = 1$ day.

(ii) Holt-Winters (HW) model (Holt, 1957):
The model implies strong correlation in demand patterns across days through a daily seasonality term s, and is expressed as:

$$\xi^z(t + h) = (l(t) + b(t) * h) * s(t - 1 + h), \tag{7}$$

where:
$l(t) = \alpha * (\xi^z(t) - s(t - 1)) + (1 - \alpha) * (l(t - 1) + b(t - 1)$,
$b(t) = \beta * (l(t) - l(t - 1)) + (1 - \beta) * b(t - 1)$,
$s(t) = \gamma * (\xi^z(t) - (l(t - 1) - b(t - 1))) + (1 - \gamma) * s(t - 1)$, and
α, β, γ are hyperparameters that are chosen using cross-validation.

(iii) Random Forest (RF) model (Breiman, 2001):
A random forest model with inputs as defined in Section 3.1. The model is of the form:

$$[\hat{\xi}^z(t + h^z + p^z), \hat{\xi}^z(t + h^z + p^z - 1), ...,\hat{\xi}^z(t + h^z)] = g(x^z(t), ...,x^z(t - p^z)), \tag{8}$$

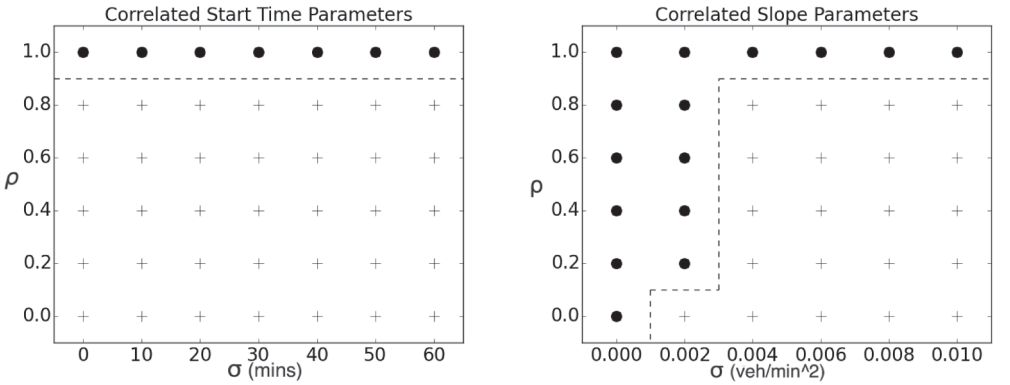
where g represents the function for the model and x^z, h^z and p^z are as defined in (5).

The idealized network layouts examined in Section 2.2 were used in five distinct tests, numbered 1–5. Each number corresponds to the scenario number tested in Section 2.2, save for the demand pattern. Demand in all five of the following tests conformed approximately to the single pulsed pattern previously shown in Fig. 3b, but rates varied as per an i.i.d. Gaussian distribution with mean 0.1 veh/min^2 , and standard deviation 0.01 veh/min^2 ; and with mean start times like those in Section 2.2, but with a standard deviation of 30 min. The daily variations in the demand rate and demand times were small, implying strong daily correlation, as typically observed in real settings.

A model for each of the five scenarios was trained on data until convergence was achieved. Congestion scores generated from simulations were treated as *ground truth*. Root Mean Squared Errors (RMSEs) of this vector from those forecasted by the deep learning and the benchmark models are presented in Table 1. Note that the deep learning model provides superior predictions as compared to the 1-NN and HW models, with differences that in all cases are either close to or exceed a factor of 2. The predictions are marginally

Table 1
RMSEs for deep learning model as compared to 1-Nearest Neighbor model, Holt-Winters model and Random Forest model.

Scenario Number as per Section 2.2	LSTM RMSE	1-NN RMSE	Holt-Winters RMSE	Random Forest RMSE
1	7.383	13.604	21.807	7.496
2	13.789	34.894	45.797	36.517
3	1.659	2.801	3.572	1.941
4	2.252	7.029	8.954	2.368
5	4.107	7.336	9.234	4.324



(a) Comparison for various (σ_{st}, ρ_{st}) pairs (b) Comparison for various (σ_{sl}, ρ_{sl}) pairs

Fig. 8. Comparison of the prediction accuracy of the deep learning model vs the 1-NN model as a function of ρ and σ when $h = p = 2$ h.

better than those provided by the RF model for all scenarios except Scenario 2, where the difference exceeds a factor of 2. The larger difference in prediction accuracy in Scenario 2 can be attributed to the presence of correlated inputs (Tolosi and Lengauer, 2011).[8] The outcomes confirm that the deep learning model is well suited to the kinds of random demand variations that can occur in real settings. We next explore how to set parameters to enhance the model's predictions for scenarios that are less suited to an LSTM model.

3.4. Tests in unfavorable scenarios

Comparisons are next drawn using the following scenarios that resemble Scenario 2 in Section 2.2; i.e., the scenarios featured time-varying ODs from multiple origins, but with added features. The features either favor the 1-NN model (i.e., the best performing benchmark model for Scenario 2), or adversely affect the LSTM model. The added features are: (i) correlation between OD demands on consecutive days; (ii) noisy input signals; and (iii) partially observed inputs.

3.4.1. Adverse feature (i): correlation between OD demands on consecutive days

As per (Ashok and Ben-Akiva, 1993; Ashok and Ben-Akiva, 2000; Zhou and Mahmassani, 2007), an auto-regressive structure was defined on the demand parameters, i.e., the start time of the demand and its rate were:

$$\begin{aligned} st(t+\Delta) &= \rho_{st} * st_t + (1-\rho_{st}) * \mu_{st} + \epsilon_{st},\ \epsilon_{st} \sim \mathcal{N}(0, \sigma_{st}) \\ sl(t+\Delta) &= \rho_{sl} * sl_t + (1-\rho_{sl}) * \mu_{sl} + \epsilon_{sl},\ \epsilon_{sl} \sim \mathcal{N}(0, \sigma_{sl}) \end{aligned} \tag{9}$$

where:

Δ is 1 day;
st_t and sl_t are the start time and rate of the demand pulse, respectively on day t,
μ_{st} and μ_{sl} are the mean start time and mean slope for the demand pulse respectively, and
ρ_{st} and ρ_{sl} are auto-correlation parameters for the start time and the rate of the pulse.

The values of μ_{st} and μ_{sl} were chosen as 7AM and 0.1 veh/min^2, respectively. Hyper-parameters h and p were tuned to achieve robustness; i.e., prediction accuracies that are superior to those of the 1-NN model for large values of ρ and small values of σ. Robustness was greatest when $h = p = 2$ h. The deep learning model is hypothesized to outperform the 1-NN model for higher values

[8] The source code for all models and comparisons are available at https://github.com/sudatta0993/Dynamic-Congestion-Prediction.

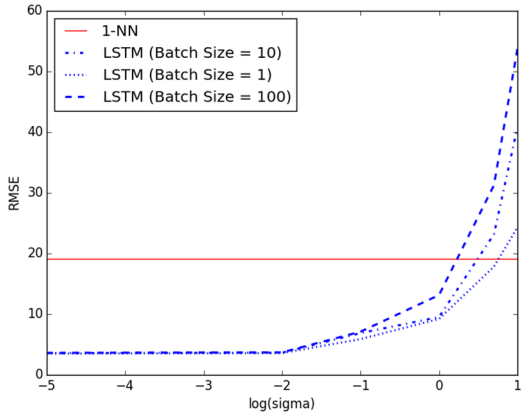


Fig. 9. Impact of measurement noise on the prediction accuracy for different batch sizes.

of σ and smaller values of ρ.

Predictions for the tuned LSTM model were compared against those of the 1-NN model for ranges of ρ and σ. Outcomes are shown in Fig. 8a and b. The "+" symbols signify instances in which the LSTM model was the more accurate of the two; and an "•" denotes the opposite outcome. The set of ρ and σ for which both models are equally accurate (an efficiency boundary) was constructed using a decision-tree classifier (Swain and Hauska, 1977); and is shown with dashed lines in the figure. Once its hyper-parameters were suitably tuned, the LSTM model frequently produced superior predictions, despite the correlations working against this.

3.4.2. Adverse feature (ii): noisy input signals

Effects of measurement noise were simulated by treating LSTM inputs as stochastic variables. For simplicity, no cross-correlation was assumed among these variables, and the relative noise in each was assumed to be equal. The following distribution was used to govern the noisy network state:

$$X_N(t) = X(t) + X(t) \odot \epsilon, \;\; \epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma), \;\; \Sigma = \sigma * I, \tag{10}$$

where:

$X(t)$ is the network state vector from (4);
$X_N(t)$ is the network state vector corrupted by measurement noise;
ϵ is a vector of random noise (assumed to be drawn from a Multivariate Gaussian distribution with mean 0 and covariance matrix Σ^z);
σ is the noise parameter of input data for each signal; and
$\odot$ represents the element-wise multiplication of vectors.

The break-even σ value (i.e., the value of σ for which the prediction accuracy of the LSTM model equals that of the 1-NN model) represents a proxy for the LSTM model's robustness.

As per Lee Giles et al. (2001), our model was made robust to measurement noise by reducing the batch-size hyper-parameter. Results are summarized in Fig. 9. On the x-axis, σ is plotted on a log scale and on the y-axis the RMSE value for the 1-NN and LSTM models are plotted for the batch sizes noted in the figure. From the plot of $log(\sigma)$, we find that the approximate break-even σ are:

- 1.8 for batch size = 100,
- 4.1 for batch size = 10, and
- 8.2 for batch size = 1.

These suggest that reducing the batch size increases the noise tolerance of the deep learning model. Hence, batch size can be tuned to preserve the model's accuracy in the event that signals are noisy.

3.4.3. Adverse feature (iii): partially observed network state

An inability to observe certain input signals in (4) was simulated by dropping some of those signals. Predictions from the LSTM model with a full complement of inputs were thereafter compared against those made with inputs missing.

Recall that in Scenario 2 of Section 2.2, vehicles from neighborhood A arrive to neighborhood B earlier than do vehicles from a second demand stream; see again Fig. 4b. Intuition suggests that arrivals from A are critical inputs to B's congestion score in the early going, say during minutes 600–720. By similar reasoning, arrivals from the second stream are probably critical inputs later in the day, say during minutes 840–960. Outcomes presented in Table 2 bear this out. Study of Table 2 reveals that dropping demand from A produced very poor predictions (i.e. high RMSE) in the early going, but had more modest impact on predictions later in the day. The

Table 2
Root Mean Squared Error for the deep learning model with omitted inputs for two different prediction horizons.

Inputs Omitted	Prediction Horizon	
	600–720 min	840–960 min
None	16.4	49.8
Demand from neighborhood A	39.0	50.4
Second demand stream	17.4	88.0

Fig. 10. Simplified freeway network of the San Francisco Bay Area. Target neighborhoods for congestion score prediction are lightly shaded.

reverse was true when the demand from the second stream was dropped.

For simple scenarios like the one studied here, intuition can be used to identify missing inputs that may be critical. The same may not be true, however, of more complex, real-world scenarios. This motivates the need for a deeper understanding of how inputs contribute to predictions. The matter will be taken up in Section 4. Before getting to that discussion, we compare LSTM to the benchmark model predictions for a larger, less-idealized network.

3.5. Tests on a more realistic network

We next compare predictions from the LSTM and the benchmark models using the network in Fig. 10. It resembles the freeway/highway system in the San Francisco Bay Area, but in a simplified form: each of 54 neighborhoods (or zones) contains a single link only.[9] Neighborhood-wide traffic can thus again be described by a single FD. Congestion scores were forecasted for the two lightly-shaded neighborhoods (labeled 1 and 2) in the figure. The two were chosen to represent varying neighborhood sizes.

The FD for each link in the network was obtained from Open Street Map of the area. Commute-time OD demands for all neighborhoods were generated using data from Census Transportation Planning Products for the years 2006–2010, and were calibrated by matching home and work locations and distributions within census tracts (U.S. Department of Transportation Federal Highway Administration, 2013). Variability in this travel demand across days was assured by drawing the start times and duration for the demand period (home-work or work-home) on each day from Gaussian distributions. Four such scenarios numbered I-IV were examined. The distributions in each scenario are summarized in Table 3. Within a day, the start time for each trip was drawn from a uniform distribution in the interval [starttime, starttime + duration]. This process was repeated to generate data for 1000 days with 100,000 persons travelling from home to work and back home on each day.

Each day's route assignment was obtained for each traveler using the micro-simulation software MATSim (Illenberger et al., 2007). Simulated OD demands, link accumulations, link travel times and trip completion rates were extracted at 5-min intervals. Additional details regarding the simulation are furnished in Appendix B.

The RMSE's were computed by comparing forecasted congestion scores against those generated by MATSim. Outcomes are presented in Table 4.[10] Note that for both neighborhoods 1 and 2, and for all four scenarios, RMSE's are smaller for the deep learning

[9] The network contains 39 nodes, since some links share end points. For more general networks, partitioning into zones must follow (Ji and Geroliminis, 2012).

Table 3
Parameter set for Gaussian distributions which determine the start time and duration of home-to-work and work-to-home trips.

Scenario	Home-to-Work trips		Work-to-Home trips	
	Standard deviation of start time[a] (h)	Standard deviation of duration[b] (h)	Standard deviation of start time[a] (h)	Standard deviation of duration[b] (h)
I	1	0	1	0
II	2	0	2	0
III	1	1	1	1.5
IV	2	1	2	1.5

[a] Mean start time = 8.5 h for home-to-work trips, and 17.5 h for work-to-home trips.
[b] Mean duration = 1 h for home-to-work trips, and 1.5 h for work-to-home trips.

Table 4
Table of RMSEs for LSTM model, Holt-Winters (HW) model, 1-Nearest Neighbor (1-NN) model, and Random Forest (RF) model for predicting congestion in neighborhoods 1 and 2.

Scenario	Neighborhood 1				Neighborhood 2			
	LSTM	1-NN	HW	RF	LSTM	1-NN	HW	RF
I	0.356	0.503	1.104	0.442	0.801	1.300	3.014	0.979
II	0.675	1.077	1.121	0.834	1.107	1.831	3.011	1.128
III	0.871	1.010	1.510	0.976	1.228	2.256	2.015	1.922
IV	0.609	0.924	1.082	0.830	2.124	3.055	2.700	2.989

model than for each of the benchmark models. The difference in RMSE values increases with increasing uncertainty, indicating that the deep learning model better accounts for daily uncertainty in demand. The test also demonstrates that the deep learning model scales well with larger networks.

4. Interpreting input effects

Tests in Section 3.4 motivate the need to identify critical inputs. A framework for doing this is offered in Section 4.1. The framework is tested in Section 4.2.

4.1. Framework for interpretation

A *Neural Attention Model*-based framework (Tang et al., 2014; Mnih et al., 2014; Show et al., 2015; Yao et al., 2015; Ramanishka et al., 2017) was used to identify the contributions of LSTM inputs to the output. The framework derives weights, also known as attentions, assigned to each input in addition to the standard neural network weights. These additional weights serve as proxies for the contributions made by those inputs in predicting the output. These attentions are learned after the LSTM is trained.

For the purpose of modeling attentions, the sequence of congestion scores forecasted by the trained model using (26) of Appendix A is considered to be the *baseline forecast*. For convenience, the forecasted congestion scores are represented by the vector $\hat{\xi}^z$ as follows:

$$\hat{\xi}^z = [\hat{\xi}^z(t+h+p), \ldots, \hat{\xi}^z(t+h)]. \tag{11}$$

The attentions corresponding to a particular input signal were derived by measuring the relative impact of omitting that signal on the forecasted congestion scores. The following equations summarize the process for calculating attentions corresponding to an input signal f that is observed l time periods (or lags) from the time of prediction, t:

[10] Details of model hyper-parameter specifications are provided in Appendix C.

(a) 600-720 mins (b) 720-840 mins (c) 840-960 mins

Fig. 11. Heatmap representing the attentions for the demand from A (left column in each plot) and the second demand stream (right column in each plot) for different prediction horizons.

$$
\begin{aligned}
\hat{\xi}^{z}_{1,f,l} &= \sum_{i=1}^{l-1} \left\{ W_{\text{LSTM}i} * \widetilde{hid}\left(t-h-i \right) + B_{\text{LSTM}i} \right\} + \sum_{i=l}^{p} \left\{ W_{\text{LSTM}i} * hid\left(t-h-i \right) + B_{\text{LSTM}i} \right\}, \\
\hat{\xi}^{z}_{2,f,l} &= \sum_{i=1}^{l} \left\{ W_{\text{LSTM}i} * \widetilde{hid}\left(t-h-i \right) + B_{\text{LSTM}i} \right\} + \sum_{i=l+1}^{p} \left\{ W_{\text{LSTM}i} * hid\left(t-h-i \right) + B_{\text{LSTM}i} \right\}, \\
\text{Loss}_{1,f,l} &= \left\| \hat{\xi}^{z} - \hat{\xi}^{z}_{1,f,l} \right\|_2, \\
\text{Loss}_{2,f,l} &= \left\| \hat{\xi}^{z} - \hat{\xi}^{z}_{2,f,l} \right\|_2, \\
\text{Loss}_{f,l} &= \text{Loss}_{1,f,l} - \text{Loss}_{2,f,l} \\
att_{f,l} &= \begin{cases} \dfrac{\text{Loss}_{f,l}}{\sum_{k \in F} \sum_{i=1}^{p} \text{Loss}_{k,i}}, & \sum_{k \in F} \sum_{i=1}^{p} \text{Loss}_{k,i} \neq 0 \\ \dfrac{1}{p * N}, & \sum_{k \in F} \sum_{i=1}^{p} \text{Loss}_{k,i} = 0 \end{cases},
\end{aligned}
\tag{12}
$$

where:

W_{LSTM} and B_{LSTM} are weight and bias vectors derived from a trained neural network based on LSTM architecture; see (26);
hid_t is the hidden state at t using (25) from Appendix A;
$\widetilde{hid}_t$ is the hidden state at t using (25) and replacing x_t with a vector where all values except input signal f are randomly permuted
$\widetilde{hid}_t$ is the hidden state at t using (25) and replacing x_t with a randomly permuted vector;
$att_{f,l}$ is the attention corresponding to input signal f at $(t-l)$;
F is the set of all inputs; and
N is the dimensionality of the input.

4.2. Testing the framework

We return to Scenario 2 in Section 2.2, and derive proxies for the contribution of each demand stream in predicting neighborhood B's congestion scores. Intuition as well as analysis conducted in Section 3.4 suggest that demand from neighborhood A affects B's congestion scores during the early going (i.e., around 600–720 min), whereas the second demand stream affects the scores during later times (i.e., around 840–960 min). Fig. 4b indicates that both demands coalesce at the entrance to neighborhood B during the time interval of 720–840 min. We now test whether attentions for the two demand streams derived by (12) reflect the same input effects as in Section 3.4. Results are summarized in Fig. 11. The figure represents the attentions for the two demand streams (represented by heatmaps in the two columns in each plot) for forecasting B's congestion score in three different time periods, namely: 600–720 min, 720–840 min, and 840–960 min. Attentions are derived for demands in 5 min intervals for the two hours prior to each prediction horizon.

For the earliest prediction horizon, 600–720 min, only demand from neighborhood A (left column) contains non-zero values. This indicates that the demand from A is the only demand that is useful for predicting congestion scores during this first interval. Similarly, only the second demand stream (right column) contains non-zero attentions for predictions between 840 and 960 min. Thus, the second demand stream is the one that substantially affects the forecasted congestion scores during this late interval. The plot for 720–840 min suggests that both demand streams are relevant for forecasting scores in this horizon, owing to coalescing demands. In summary, the attentions derived by the Neural Attention-based framework support our earlier intuition regarding the contribution of individual inputs to congestion scores. This is evidence that an Attention-based framework can be used to quantify input effects in complex scenarios where intuition alone may fail.

5. Model improvements via graphical input representation

A framework combining Convolutional Neural Networks (CNNs) (Fukushima, 1987; LeCun et al., 1998) with LSTM was developed to predict traffic states more accurately. Input information was encoded using spectral graph theory (Bruna et al., 2013; Defferrard et al., 2016; Kipf and Welling, 2016) to allow learning relationships from graphical representation of inputs.

Consider a graph $G = (\mathscr{V}, \mathscr{E}, W)$, where $\mathscr{V}$ is the set of nodes ($|\mathscr{V}| = n$), $\mathscr{E}$ is the set of edges and $W \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix encoding the connection weight between any two vertices. The graph Laplacian matrix is denoted by L, which was diagonalized using the Fourier basis U as $L = U\Lambda U^T$, where $\Lambda = diag(\lambda_0, ..., \lambda_{n-1})$ (Shuman et al., 2013). Input signal x was filtered by a graph $g(\theta)$ to produce an output y as follows:

$$y = g_\theta(L)x = g_\theta(U\Lambda U^T)x = Ug_\theta(\Lambda)U^T x, \tag{13}$$

where, $U^T x$ is the Graph Fourier Transform.

For ease of computation, the function $g_\theta(\Lambda)$ was approximated through Chebychev polynomials (Defferrard et al., 2016). The pooling operations in CNNs were extended to graphical inputs through agglomerative clustering in a neighborhood around graph nodes (Bruna et al., 2013).

5.1. Graphical representation of inputs

Consider a graph $G = (\mathscr{V}, \mathscr{A})$ representing the road network, where $\mathscr{A}$ represents directed road links approximated by straight line segments and $\mathscr{V}$ represents end points, with Z neighborhoods spanning the area covered by G. Construct weighted, undirected graphs $\widetilde{G} = (\widetilde{\mathscr{V}}, \widetilde{\mathscr{E}}, W)$ based on the relationships between inputs in (4). Weights are determined based on distance measurements between OD pairs in a transformed vector space. The distance is a function of similarities in the distribution of route choices for trips belonging to each OD pair. The steps are described next.

Each $\tilde{v} \in \widetilde{\mathscr{V}}$ represents a pair of neighborhoods $(z_\alpha, z_\beta) \in Z$. Input signals in (4) are transformed into input signals for $\widetilde{G}$ through the following transformations:

- The OD demand, D^{z_α, z_β}, is assigned to node v_a representing OD pair (z_α, z_β).
- Each link count, C_i on link i, is distributed based on prior probabilities of origin and destination zones of all trips observed on link i; see (14).
- Each link travel time, TT_i on link i, is likewise distributed based on prior probabilities of origin and destination zones of all trips observed on link i; see again (14).

Therefore, the transformed input signal for node v_a to predict macroscopic congestion in neighborhood z_β is be calculated as:

$$\widetilde{x}_a^{z_\beta}(t) = \begin{bmatrix} D^{z_\alpha, z_\beta}(t) \\ \sum_i p\left(\alpha, \beta \middle| i\right) C_i(t) \\ \sum_i p\left(\alpha, \beta \middle| i\right) TT_i(t) \end{bmatrix}, \tag{14}$$

where $p(\alpha, \beta|i)$ is the prior probability of a trip observed on link i having an OD pair as (z_α, z_β).

All trips in the region are clustered based on the similarities of the sequence of links traversed using *Dynamic Time Warping* (DTW) (Berndt and Clifford, 1994). For each OD pair (z_α, z_β), the probability distribution over derived clusters, $\Delta^{z_\alpha, z_\beta}$, is determined as the proportion of trips with origin z_α and destination z_β belonging to each cluster. Consider nodes v_a, v_b representing the pairs of zones (z_α, z_β) and (z_γ, z_δ), respectively. The weight between nodes $v_a, v_b \in \widetilde{\mathscr{V}}$ is calculated as:

$$W_{ab} = D_{KL}(\Delta^{z_\alpha, z_\beta}, \Delta^{z_\gamma, z_\delta}), \tag{15}$$

where D_{KL} represents the K-L divergence between two distributions.

Table 5
Table describing RMSE comparison for Graph-CNN + LSTM model and LSTM model for predicting congestion in neighborhoods 1 and 2.

Scenario	Neighborhood 1		Neighborhood 2	
	LSTM	Graph-CNN + LSTM	LSTM	Graph-CNN + LSTM
I	0.356	0.187	0.801	0.596
II	0.875	0.358	1.077	0.558
III	0.871	0.102	1.228	0.435
IV	0.609	0.194	2.124	0.858

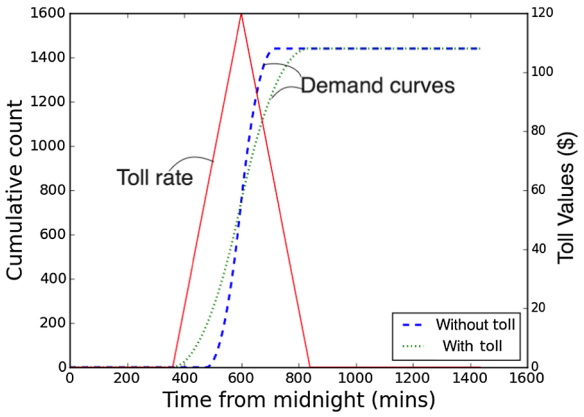


Fig. 12. Cumulative departure curve (primary y-axis) under pulse-shaped tolling and toll rate (secondary y-axis).

5.2. Model testing

We compare predictions of the Graph-CNN + LSTM model described in Section 5.1 against those produced by an LSTM model without graphical input representation. For ease of computation, the graph was made sparser by restricting the connections for each node to its 5 nearest neighbors only. Table 5 presents the comparison of the prediction accuracy for the two models.

We observe that the prediction accuracy for the Graph CNN + LSTM models is better than that of the LSTM-only model. The gain in performance may be attributed to better feature learning by exploiting the graphical nature of the input data.

6. Application

We now show by example how forecasted congestion scores and suitable interpretations of input effects can enhance congestion-management strategies. The example entails a simple congestion-pricing scheme in which commuters are tolled as they depart from their origins, after having input their destinations (e.g. via an app). Each commuter's toll amount is based on the predicted impact of her trip on the congestion score for a neighborhood z; and is aimed at deterring commuters from arriving at z during the peak.

Assumptions regarding commuters' response to tolls are presented in Section 6.1. The objective function is defined and solved in Section 6.2. The solution requires estimation of: parameters for the marginal impact of each arrival at z on its congestion score; the probability of a trip destination in z; and a constant of proportionality to convert the solution to a dollar amount. These parameter estimates are separately obtained by the LSTM model in Section 6.3, and by estimating future demands in Section 6.4. The tolls generated by these two methods, and the effectiveness of these tolls, are compared in Section 6.5.

6.1. Behavioral assumptions under tolling

Assume that all commuters value their time at fixed rate \$30/h. Further assume that: the toll varies with time (i.e. pulses) as per the solid lines shown in Fig. 12; and the time varying charge is $\in$ greater than \$30/h. Commuters are assumed to behave as per the work in Arnott et al. (1993), so as to produce network-wide equilibrium as per (Vickrey, 1969). The dashed curve in Fig. 12 displays cumulative departures that occur in the absence of tolling. Departures in response to the toll would be as displayed by the dotted curve. To see why this is true, we note that the first departing commuter unilaterally improves her lot by departing earlier. Her utility increases at progressively earlier times, until reaching a departure time that coincides with the start of tolling. She therefore departs at that start time. The last departing commuter similarly improves her situation by departing later, and ultimately chooses to do so at the toll's ending time. The toll rate momentarily flattens at its peak, such that the commuter who departs at that time does so with or without tolling.

6.2. Optimal toll conditions

Assume that a trip departing neighborhood z_i at time t_i arrives at the periphery of neighborhood z at time t. We define an objective function to minimize z's cumulative congestion score while maintaining each OD demand:

$$\underset{D^{z_i,z}(t)}{Minimize} \int_{t_{start}}^{t_{end}} \xi^{z}(t)dt$$
$$s.\ t. \int_{t_{start}}^{t_{end}} D^{z_i,z}(t)dt = Constant,\ \ \forall\ i \in \left\{1, 2, ..., Z\right\}, \tag{16}$$

where:

$\xi^z(t)$ is the congestion score at t, which is a function of the demands from all Z neighborhoods;
$D^{z_i,z}(t)$ is the demand from z_i to z at t; and
t_{start}, t_{end} are the start and end times of the analysis period, respectively.

The optimization problem (16) is non-convex due to the objective function and its constraint. We achieve a lower bound on the optimal objective by solving the Lagrangian dual version of the problem. From simulations, we find that the lower bound is close to the optimum objective. The Lagrangian is formed as follows:

$$\begin{aligned}\mathscr{L} &= \int_{t_{start}}^{t_{end}} \xi^z(t)dt + \sum_{i\in\{1,2,\ldots,Z\}} \nu_i\left(\int_{t_{start}}^{t_{end}} D^{z_i,z}(t)dt - Constant\right)\\ &= \int_{t_{start}}^{t_{end}} \left\{\xi^z(t) + \sum_{i\in\{1,2,\ldots,Z\}} \nu_i\left(D^{z_i,z}(t) - Constant\right)\right\}dt.\end{aligned} \tag{17}$$

The first order condition to minimize the Lagrangian function gives the following condition:

$$\begin{aligned}&\frac{\partial \mathscr{L}}{\partial D^{z_i,z}(t)} = \int_{t_{start}}^{t_{end}} \left\{\frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)} + \nu_i\right\}dt = 0\\ \Rightarrow\ &\int_{t_{start}}^{t_{end}} \frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)}dt = Constant,\ \forall\ i \in \{1, 2, \ldots, Z\},\end{aligned} \tag{18}$$

where:
$\frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)}$ is the marginal impact of demand from z_i on z's congestion score at t.
The optimality condition can be interpreted as each demand stream equally impacting the congestion score over the analysis period. For ease of computation, we impose a stricter condition in which the impact of all demands on the congestion score is constant over both space and time. Further assume that the marginal impact of each demand on the congestion score scales linearly with the toll. The optimal toll rate is therefore:

$$Toll_{z_i,t_i} = k_{z_i,t}\left(t_i,\ t\right) * \frac{\partial \xi^z(D^{\cdot,z}(t))}{\partial D^{z_i,z}(t)}, \tag{19}$$

where:
$k_{z_i,z}(t_i,\ t)$ is a constant of proportionality to convert the expression to a dollar amount.
The term $\frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)}$ is impacted by two quantities:

(i) The probability that a trip departing from z_i at t_i arrives at z at t, henceforth denoted $p_{z_i,z}(t_i,\ t)$; and
(ii) the marginal impact of each arrival at z at t on the congestion score, henceforth denoted $\beta^{imp,z}(t)$.

Both quantities are assumed to be independent and to affect $\frac{\partial \xi^z(t)}{\partial D^{z_i,z}(t)}$ linearly, which gives:

$$\frac{\partial \xi^z(D^{\cdot,z}(t))}{\partial D^{z_i,z}(t)} = p_{z_i,z}\left(t_i,\ t\right) * \beta^{imp,z}(t). \tag{20}$$

The $\beta^{imp,z}(t)$ is derived in terms of observable quantities (see Section 2.1) as follows:

$$\begin{aligned}\beta^{imp,z}(t) &= \frac{\partial \xi^z(t)}{\partial \sum_{i\in A^z}(A_i(t))}\\ &= \frac{\partial \xi^z(t)\,/\,\partial t}{\partial \sum_{i\in A^z}(A_i(t))\,/\,\partial t}\\ &= \frac{\beta_{\xi,z}(t)}{\partial \sum_{i\in A^z}(A_i(t))\,/\,\partial t}\\ &= \frac{\beta_{\xi,z}(t)}{\partial \sum_{i\in A^z} n_i(t)\,/\,\partial t + \partial(\sum_{i\in A^z} L_i(t))\,/\,\partial t}\\ &\overset{(a)}{\approx} \frac{\beta_{\xi,z}(t)}{\partial \sum_{i\in A^z} n_i(t)\,/\,\partial t + \partial(\sum_{i\in A^z} E_i(t))\,/\,\partial t}\\ &= \frac{\beta_{\xi,z}(t)}{\partial \sum_{i\in A^z} n_i(t)\,/\,\partial t + T^z(t)}\\ &= \frac{\beta_{\xi,z}(t)}{\beta^{n,z}(t) + T^z(t)},\end{aligned} \tag{21}$$

where:

$\beta_{\xi,z}(t)$ is the rate of change in z's congestion score at t, and
$\beta^{n,z}(t)$ is the rate of change in z's accumulation at t.

Condition (a) follows from the assumption that the number of trips that start outside z but terminate inside z are either nearly constant over time, or are very small compared to the number of trips with start and end locations within z.

Combining (19)–(21) yields the following condition for the optimal congestion toll:

$$Toll_{z_i,t_i} = k_{z_i,t}\left(t_i,\, t\right) * p_{z_i,z}\left(t_i,\, t\right) * \frac{\beta_{s,z}(t)}{\beta^{n,z}(t) + T^{z}(t)}. \tag{22}$$

6.3. Optimal toll using LSTM model

The optimal toll in (22) was next approximated by an LSTM model as follows:

(i) The expression for predicting $\beta^{imp,z}(t)$ was approximated by model predictions:

$$[\beta^{imp,z}(t+h+p),\ ...,\beta^{imp,z}(t+h)] \approx W_{\text{LSTM}}^{T}[hid_{t-h},\ ...,hid_{t-h-p}] + B_{\text{LSTM}}. \tag{23}$$

Ground truth values were calculated from (20) and the input to the LSTM model was the network state vector in (4).

(ii) The expression for predicting $p_{z_i,z}(t_i,\, t)$ was approximated by attention values for the demand from z_i (see (12)):

$$p_{z_i,z}\left(t_i,\, t\right) \approx att_{f^*,l} = \begin{cases} \dfrac{\text{Loss}_{f^*,l}}{\sum_{k\in F}\sum_{i=1}^{p}\text{Loss}_{k,i}}, & \sum_{k\in F}\sum_{i=1}^{p}\text{Loss}_{k,i} \neq 0 \\ \dfrac{1}{p*N}, & \sum_{k\in F}\sum_{i=1}^{p}\text{Loss}_{k,i} = 0 \end{cases}, \tag{24}$$

where:

f^* is the demand from z_i to z.

(iii) $k_{z_i,z}(t_i,\, t)$ was trained using cross-validation, and was treated in the present work as constant over time, so as to limit the number of iterations needed for training.

6.4. Optimal toll using average past demands

The optimal toll in (22) was approximated with the assumption that future demand patterns mimic the average demand rates and times from past days. It was further assumed that demands follow nearly-identical pulsed patterns, and coalesce at z for a very short duration.

Consider n pulsed demands with nearly-identical rates, such that the i^{th} average pulsed demand starts and ends at $t_{i,s}$ and $t_{i,e}$, respectively. Assume that $t_{i,s} < t_{i+1,s}$ and $|t_{i,e} - t_{i+1,s}| < \delta,\ \forall\ i \in \{1,\, 2,\, ...,n-1\}$. The start and end times for the analysis period are t_{start} ($< t_{1,s}$) and t_{end} ($> t_{n,e}$), respectively. In this scenario, an approximately optimal toll is:

(a) monotonically increasing between $[t_{start},\ (t_{1,e} + t_{2,s})/2]$ for departures in the 1^{st} demand pulse;
(b) monotonically decreasing between $[(t_{n-1,e} + t_{n,s})/2,\ t_{end}]$ for departures in the n^{th} demand pulse; and
(c) pulse-shaped curve between $[(t_{i-1,e} + t_{i,s})/2,\ (t_{i,e} + t_{i+1,s})/2]$ for departures in any other demand pulse i.

6.5. Comparisons

Simulations were conducted for scenario 2 in Section 2.2 with small daily variations in demand that were in line with those described in Section 3.3. Cumulative congestion scores for neighborhood B, average travel times inside neighborhood B, and average tolls were separately obtained for the tolling schemes generated by the LSTM in Section 6.3, and for average demands as in Section 6.4. Outcomes are presented in Table 6, along with a baseline (no-tolling) scenario. Tabulated values are the averages of 100 simulations. The outcomes indicate that both tolling schemes do a good job of reducing cumulative congestion scores and travel times, with the LSTM model performing slightly better. Importantly, only the LSTM model produced a reasonable toll of $7.96, as compared against the politically-untenable alternate toll of nearly $50 on average. The lower toll value was produced by the LSTM model because it allowed the pulsed demands to coalesce in neighborhood B when those rates were low. Contrary to the alternative strategy, excessively high tolls were not needed to coerce commuters into avoiding the rush by traveling very early or very late in the day. Note that avoiding this coercion is a further advantage of the LSTM approach.

Table 6
Outcomes for various toll strategies.

Scenario	Cum. congestion score (min^2)	Average travel time (mins/veh)	Average Toll ($/veh)
No Toll	81,859	283.13	0.0
Average demand-based toll	21,525	46.84	48.32
LSTM model-based toll	19,428	44.63	7.96

7. Conclusions

The present work indicates that a deep learning model based on LSTM architecture can play useful roles in combating neighborhood traffic congestion inside a larger region. Simulated tests show that the forecasted output, neighborhood congestion scores, is sensitive to regional traffic conditions, and can be predicted from four real-time signals that serve as proxies for those conditions. Predictions were shown to be superior to those produced by three benchmark models. This was true for very simple networks as well as for a more realistic one; and tended to hold even in unfavorable settings. Improvements to the model were made by representing inputs through weighted undirected graphs. The proposed Graph-CNN + LSTM model bettered the prediction accuracy of our LSTM model. A Neural Attention Model was shown to be effective in approximating the contributions of individual inputs to forecasted scores, which helps identify the potential cause of congestion. Finally, the model was shown to be helpful in devising a congestion tolling scheme.

In light of the above, we believe that the present model can be used both for managing a region's recurrent (eg. daily or seasonal) congestion problems, and for those induced by large incidents, whether foreseeable or otherwise. Moreover, the work opens a door to combating congestion over large, possibly multi-jurisdictional spatial scales. Its data-driven character may prove useful in sharing inputs across distinct jurisdictions, and in implementing control measures that serve entire regions.

Appendix A. Steps executed by a typical Long Short-Term Memory (LSTM) model

A typical LSTM model (Hochreiter and Schmidhuber, 1997) undertakes the following operations at each time step:

$$\begin{aligned} fg_t &= \sigma(W_f^T [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i^T [h_{t-1}, x_t] + b_i) \\ o_t &= \sigma(W_o^T [h_{t-1}, x_t] + b_o) \\ c_t &= fg_t \odot c_{t-1} + i_t \odot tanh(W_c^T [h_{t-1}, x_t] + b_c) \\ hid_t &= o_t \odot tanh(c_t) \end{aligned} \tag{25}$$

where:

fg_t, i_t, o_t, c_t and hid_t are the forget, input, output, cell-state and hidden states at time t, respectively;
W_f, W_i, W_o and W_c are the weight vectors for the forget, input, output and cell-state gates;
b_f, b_i, b_o and b_c are the bias vectors for the forget, input, output and cell-state gates;
σ and $tanh$ represents the Sigmoid and Hyperbolic Tangent Activation Functions; and
$\odot$ represents the Hadamard product between two vectors.

If the model implements stacked LSTM layers, then the final hidden layer is calculated iteratively from the previous hidden layers with usual weights, biases and activation functions.

For the model proposed in (5), the final output can be expressed in terms of the input as:

Table 7
Hyper-parameter set for LSTM model structure and training process for scenarios described in Section 2.2 and demand patterns across multiple days, as described in Section 3.3

Scenario Number	Inputs	Number of hidden layers	Number of iterations
1	$D(t)$	500	150
2	$D(t)$	100	400
3	$D(t)$, $TT(t)$, $C(t)$	100	150
4	$D(t)$, $TT(t)$, $C(t)$	100	200
5	$TT(t)$, $C(t)$, $\xi^B(t)$	100	150

In all the above experiments:
h^z = 2 h
p^z = 24 h
Number of stacked LSTM layers = 1
Learning rate = 0.01
Batch size = 10

$$\begin{aligned}\hat{\xi}^{z}(t) &= W_{LSTM}^{T}[hid_{t-h}, hid_{t-h-1}, \ldots, hid_{t-h-p}] + B_{LSTM} \\ &= \sum_{i=0}^{p}\left\{W_{LSTMi} * hid\left(t-h-i\right) + B_{LSTMi}\right\},\end{aligned} \tag{26}$$

where:

W_{LSTM} and B_{LSTM} are weight and bias vectors for the final prediction.

A.1. LSTM model structure and hyper-parameter values for sceanrios described in Section 2.2

Table 7

Appendix B. Additional experimental setup details for Section 3.5

B.1. Network generation

The full-scale network was extracted from Open Street Maps (OSM) using open source Java application OSMOSIS.[11] Features of the freeways and highways were further extracted by querying the OSM data and filtering over the road properties, including the road type, the number of lanes and the speed limit. Paths continuing along a road were represented by a single link to further simplify the network. The resulting network was comprised of 39 nodes and 54 links. For the purpose of this example, the Traffic Analysis Zones (TAZs) were merged and grouped depending on the closest link in the simplified network. Hence, the analysis scenario included 54 aggregated TAZs.

B.2. Demand generation

The demand generation process involved the following steps:

1. *OD table generation*:
   The overall OD demand was generated from Census Transportation Planning Products data by first extracting the aggregate employed adults (16 years of age and over) within census tracts, and their workplace distribution.
2. *Initial commute activity chain generation*:
   The population size was fixed at 100,000. Each individual's commute activity chain was home → work → home. The departure time for each trip was fixed in the interval [starttime, starttime + standarddeviation] with parameter values specified in Table 3. This process was repeated for each individual for 1000 days.
3. *Final route assignment*:
   The daily equilibrium traffic flow was produced using the micro-simulation software MATSim (Horni and Nagel, 2016).

Table 8
Parameter set for each LSTM model developed for simplified Bay Area freeway network.

Scenario Number	Neighborhood for prediction	Number of iterations
1	1	200
1	2	200
2	1	300
2	2	300
3	1	300
3	2	300
4	1	400
4	2	600

In all above experiments:
Model inputs are: $D(t)$, $TT(t)$ and $C(t)$.
$h^{z} = 0$.
$p^{z} = 24$ h
Number of hidden layers = 100
Number of stacked LSTM layers = 10
Learning rate = 0.001
Batch size = 10

[11] http://wiki.openstreetmap.org/wiki/Osmosis.

Appendix C. LSTM model structure and hyper-parameter values for SF bay area network and scenarios described in Table 3

Table 8.

References

Aboudolas, Konstantinos, Geroliminis, Nikolas, 2013. Perimeter and boundary flow control in multi-reservoir heterogeneous networks. Transp. Res. Part B: Methodol. 55, 265–281.

Aboudolas, Konstantinos, Papageorgiou, Markos, Kosmatopoulos, E., 2009. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. Transp. Res. Part C: Emerg. Technol. 17 (2), 163–174.

Arel, Itamar, Rose, Derek C., Karnowski, Thomas P., et al., 2010. Deep machine learning-a new frontier in artificial intelligence research. IEEE Comput. Intell. Mag. 5 (4), 13–18.

Arnott, Richard, Small, Kenneth, 1994. The economics of traffic congestion. Am. Scientist 446–455.

Arnott, Richard, De Palma, Andre, Lindsey, Robin, 1993. A structural model of peak-period congestion: A traffic bottleneck with elastic demand. Am. Econ. Rev. 161–179.

Ashok, Kalidas, Ben-Akiva, Moshe E., 1993. Dynamic origin-destination matrix estimation and prediction for real-time traffic management systems. In: International Symposium on the Theory of Traffic Flow and Transportation (12th: 1993: Berkeley, Calif.). Transportation and traffic theory.

Ashok, Kalidas, Ben-Akiva, Moshe E., 2000. Alternative approaches for real-time estimation and prediction of time-dependent origin-destination flows. Transp. Sci. 34 (1), 21–36.

Berndt, Donald J., Clifford, James, 1994. Using dynamic time warping to find patterns in time series. In: KDD Workshop, vol. 10. 16. Seattle, WA. 1994, pp. 359–370.

Breiman, Leo, 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Bruna Joan, et al., 2013. Spectral networks and locally connected networks on graphs. In: arXiv preprint arXw:1312.6203.

Ceder, Avishai, 1982. Relationships between road accidents and hourly traffic flow—II: Probabilistic approach. Accident Anal. Prevent. 14 (1), 35–44.

Chang, Gang-Len, Jileng, Wu., 1994. Recursive estimation ol time-varying origin-destination flows from traffic counts in freeway corridors. Transp. Res. Part B: Methodol. 28 (2), 141–160.

Cleveland, William S., 1981. LOWESS: A program for smoothing scatterplots by robust locally weighted regression. Am. Statistician 35 (1), 54.

Daganzo, Carlos F, 2001. A simple traffic analysis procedure. Netw. Spatial Econ. 1 (1-2), 77–101.

Daganzo, Carlos F, 2007. Urban gridlock: Macroscopic modeling and mitigation approaches. Transp. Res. Part B: Methodol. 41 (1), 49–62.

Daganzo, Carlos F, Geroliminis, Nikolas, 2008. An analytical approximation for the macroscopic fundamental diagram of urban traffic. Transp. Res. Part B: Methodol. 42 (9), 771–781.

Daganzo, Carlos F., Gayah, Vikash V., Gonzales, Eric J., 2012. The potential of parsimonious models for understanding large scale transportation systems and answering big picture questions. EURO J. Transp. Logist. 1 (1-2), 47–65.

Defferrard, Michael, Bresson, Xavier, Vandergheynst, Pierre, 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, pp. 3844–3852.

Diakaki, Christina, Papageorgiou, Markos, Aboudolas, Kostas, 2002. A multivariable regulator approach to traffic-responsive network-wide signal control. Control Eng. Pract. 10 (2), 183–195.

Edie, L.C., 1965. Discussion of traffic stream measurement and description. In: Almond, J. (Ed.), Proc. Second Intern. Symp. on the Theory of Road Traffic Flow. OECD, Paris.

Fukushima, Kunihiko, 1987. Neural network model for selective attention in visual pattern recognition and associative recall. Appl. Opt. 26 (23), 4985–4992.

Gayah, Vikash V., Gao, Xueyu Shirley, Nagle, Andrew S., 2014. On the impacts ol locally adaptive signal control on urban network stability and the macroscopic lundamental diagram. Transp. Res. Part B: Methodol. 70, 255–268.

Geroliminis, Nikolas, Daganzo, Carlos F, 2008. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. Transp. Res. Part B: Methodol. 42 (9), 759–770.

Geroliminis, Nikolas, Levinson, David M., 2009. Cordon pricing consistent with the physics of overcrowding. In: Transportation and Traffic Theory 2009: Golden Jubilee. Springer, pp. 219–240.

Geroliminis, Nikolas, Haddad, Jack, Ramezani, Mohsen, 2013. Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach. IEEE Trans. Intell. Transp. Syst. 14 (1), 348–359.

Girault, Jan-Torben, et al., 2016. Exploratory analysis ol signal coordination impacts on macroscopic lundamental diagram. Transp. Res. Rec.: J. Transp. Res. Board 2560, 36–46.

Golob, Thomas F., Recker, Wilfred W., 2003. Relationships among urban freeway accidents, traffic flow, weather, and lighting conditions. J. Transp. Eng. 129 (4), 342–353.

Gonzales, Eric J., Daganzo, Carlos F., 2012. Morning commute with competing modes and distributed demand: user equilibrium, system optimum, and pricing. Transp. Res. Part B: Methodol. 46 (10), 1519–1534.

Haddad, Jack, Geroliminis, Nikolas, 2012. On the stability of traffic perimeter control in two-region urban cities. Transp. Res. Part B: Methodol. 46 (9), 1159–1176.

Haddad, Jack, Mirkin, Boris, 2016. Adaptive perimeter traffic control ol urban road networks based on MFD model with time delays. Int. J. Robust Nonlinear Control 26 (6), 1267–1285.

Hajiahmadi, Mohammad, et al., 2013. Optimal dynamic route guidance: A model predictive approach using the macroscopic lundamental diagram. In: Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on. IEEE, pp. 1022–1028.

Hajiahmadi, Mohammad, et al., 2015. Optimal hybrid perimeter and switching plans control for urban traffic networks. IEEE Trans. Control Syst. Technol. 23 (2), 464–478.

Hale, D., 2005. Traffic network study tool: TRANSYT-7F, United States version. In: Me-Trans Center in the University of Florida.

Hochreiter, Sepp, Schmidhuber, Jiirgen, 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

CC Holt. Forecasting trends and seasonals by exponentially weighted averages. Carnegie Institute of Technology. Tech. rep. Pittsburgh ONR memorandum, 1957.

Horni, Andreas, Nagel, Kai, Axhausen, Kay W., 2016. The Multi-agent Transport Simulation MAT-Sim. Ubiquity Press, London.

Illenberger, Johannes, Flotterod, Gunnar, Nagel, Kai, 2017. Enhancing matsim with capabilities of within-day re-planning. In: Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE. IEEE. 2007, pp. 94–99.

Ji, Yuxuan, Geroliminis, Nikolas, 2012. On the spatial partitioning of urban transportation networks. Transp. Res. Part B: Methodol. 46 (10), 1639–1656.

Jusoh, Ruzanna Mat, Ampountolas, Konstantinos, 2017. Multi-gated perimeter flow control ol transport networks. In: Control and Automation (MED), 2017 25th Mediterranean Conference on. IEEE, pp. 731–736.

Keyvan-Ekbatani, Mehdi, et al., 2012. Exploiting the fundamental diagram of urban networks for feedback-based gating. Transp. Res. Part B: Methodol. 46 (10), 1393–1403.

Keyvan-Ekbatani, Mehdi, et al., 2016. Queuing under perimeter control: Analysis and control strategy. In: Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on. IEEE, pp. 1502–1507.

Kipf, Thomas N., Welling, Max, 2016. Semi-supervised classification with graph convolutional networks. In: arXiv preprint arXiv.1609.02907.

Knoop, Victor L., Hoogendoorn, S.P., Van Lint, J.W.C., 2013. The impact ol traffic dynamics on macroscopic lundamental diagram. In: 92nd Annual Meeting Transportation Research Board, Washington, USA, 13–17 January 2013; Authors version. Transportation Research Board. 2013.

LeCun, Yann, et al., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.
Lee Giles, C., Lawrence, Steve, Tsoi, Ah Chung, 2001. Noisy time series prediction using recurrent neural networks and grammatical inference. In: Machine learning 44. 1, pp. 161–183.
Lin, Sh.u., et al., 2012. Efficient network-wide model-based predictive control for urban traffic networks. Transp. Res. Part C: Emerg. Technol. 24, 122–140.
Mnih, Volodymyr, Heess, Nicolas, Graves, Alex, et al., 2014. Recurrent models of visual attention. In: Advances in Neural Information Processing Systems, pp. 2204–2212.
Mohanty, Sudatta, Pozdnukhov, Alexey, 2019. Dynamic origin-destination demand estimation from link counts, cellular data and travel time data. In: Proc. 15th World Conference on Transport Research (WCTR). Mumbai, India.
Newell, Gordon F., 1993. A simplified theory of kinematic waves in highway traffic, part I: General theory. Transp. Res. Part B: Methodol. 27A, 281–287.
Ni, Wei, Cassidy, Michael, 2018. City-wide traffic control: modeling impacts ol cordon queues.
Okutani, Iwao, Stephanedes, Yorgos J., 1984. Dynamic prediction ol traffic volume through Kalman filtering theory. Transp. Res. Part B: Methodol. 18 (1), 1–11.
Ortigosa, Javier, Menendez, Monica, Gayah, Vikash V., 2015. Analysis ol network exit lunctions for various urban grid network configurations. Transp. Res. Rec.: J. Transp. Res. Board 2491, 12–21.
Papageorgiou, Markos, et al., 2003. Review of road traffic control strategies. Proc. IEEE 91 (12), 2043–2067.
Ramanishka, Vasili, et al., 2017. Top-down visual saliency guided by captions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
Ramezani, Mohsen, Haddad, Jack, Geroliminis, Nikolas, 2015. Dynamics ol heterogeneity in urban networks: aggregated traffic modeling and hierarchical control. Transp. Res. Part B: Methodol. 74, 1–19.
Robertson, Dennis I., David Bretherton, R., 1991. Optimizing networks of traffic signals in real time-the SCOOT method. IEEE Trans. Veh. Technol. 40 (1), 11–15.
Show, Kelvin Xu, et al., 2015. attend and tell: Neural image caption generation with visual attention. In: International Conference on Machine Learning, pp. 2048–2057.
Shuman, David I., et al., 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag. 30 (3), 83–98.
Sutton, Richard S., Barto, Andrew G., et al., 1998. Reinforcement Learning: An Introduction. MIT Press.
Swain, Philip H., Hauska, Hans, 1977. The decision tree classifier: Design and potential. IEEE Trans. Geosci. Electron. 15 (3), 142–147.
Tang, Yichuan, Srivastava, Nitish, Salakhutdinov, Ruslan R., 2014. Learning generative models with visual attention. In: Advances in Neural Information Processing Systems, pp. 1808–1816.
Tolosi, Laura, Lengauer, Thomas, 2011. Classification with correlated features: unreliability of feature ranking and solutions. Bioinformatics 27 (14), 1986–1994.
U.S. Department of Transportation Federal Highway Administration, CTPP 2006–2010 Census Tract Flows. In: (2013). URL: http://www.7o20fhwa.dot.gov/planning/census_issues/ctpp/data_products/2006-2010_tract_flows/index.cfm.
Vickrey, William S., 1969. Congestion Theory and Transport Investment. English. In: The American Economic Review 59.2, pp. 251–260. ISSN: 00028282. URL: http://www.jstor.org/stable/1823678.
Wardrop, John Glen, Whitehead, J.I., 1952. Some theoretical aspects of road traffic research. Proc. Inst. Civ. Eng. 1 (5), 767–768.
Yang, Xianfeng, Yang, Lu., Hao, Wei, 2017. Origin-destination estimation using probe vehicle trajectory and link counts. J. Adv. Transp. 2017.
Yao, Li, et al., 2015. Describing videos by exploiting temporal structure. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4507–4515.
Yildirimoglu, Mehmet, Geroliminis, Nikolas, 2013. Experienced travel time prediction for congested freeways. Transp. Res. Part B: Methodol. 53, 45–63.
Zheng, Nan, et al., 2012. A dynamic cordon pricing scheme combining the Macroscopic Fundamental Diagram and an agent-based traffic model. Transp. Res. Part A: Policy Pract. 46 (8), 1291–1303.
Zheng, Weizhong, Lee, Der-Horng, Shi, Qixin, 2006. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. J. Transp. Eng. 132 (2), 114–121.
Zhou, Xuesong, Mahmassani, Hani S, 2007. A structural state space model for real-time traffic origin-destination demand estimation and prediction in a day-to-day learning framework. Transp. Res. Part B: Methodol. 41 (8), 823–840.