

Learning to Recommend Signal Plans under Incidents with Real-Time Traffic Prediction

Weiran Yao¹ and Sean Qian^{1,2}

Transportation Research Record
2020, Vol. 2674(6) 45–59
© National Academy of Sciences:
Transportation Research Board 2020
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0361198120917668
journals.sagepub.com/home/trr
 SAGE

Abstract

The main question to address in this paper is to recommend optimal signal timing plans in real time under incidents by incorporating domain knowledge developed with the traffic signal timing plans tuned for possible incidents, and learning from historical data of both traffic and implemented signals timing. The effectiveness of traffic incident management is often limited by the late response time and excessive workload of traffic operators. This paper proposes a novel decision-making framework that learns from both data and domain knowledge to real-time recommend contingency signal plans that accommodate non-recurrent traffic, with the outputs from real-time traffic prediction at least 30 min in advance. Specifically, considering the rare occurrences of engagement of contingency signal plans for incidents, it is proposed to decompose the end-to-end recommendation task into two hierarchical models—real-time traffic prediction and plan association. The connections between the two models are learnt through metric learning, which reinforces partial-order preferences observed from historical signal engagement records. The effectiveness of this approach is demonstrated by testing this framework on the traffic network in Cranberry Township, Pennsylvania, U.S., in 2019. Results show that the recommendation system has a precision score of 96.75% and recall of 87.5% on the testing plan, and makes recommendations an average of 22.5 min lead time ahead of Waze alerts. The results suggest that this framework is capable of giving traffic operators a significant time window to access the conditions and respond appropriately.

Most traffic management centers (TMC) operate a coordinated signal system that relies on historically generated signal timings, coupled with real-time technology, to manage day-to-day operations on the local network. Unfortunately, any planned or unplanned incidents (e.g., hazardous weather conditions, accidents, local events, etc.) on the network can cause catastrophic traffic gridlocks. According to the Federal Highway Administration (FHWA) in 2019, about half of congestion is nonrecurring, among which 25% is caused by accidents, 15% by weather, and 10% by work zones (1). To keep traffic flowing during these occurrences, local TMCs develop incident timing plans, or contingency traffic plans, for their owned signalized intersections to manage incident-induced congestion. However, the effectiveness of existing incident management is often limited by late response time and excessive workload of traffic operators, and the main causes are two-fold: (i) the lack of real-time and advance awareness of road conditions: traffic operators often react after receiving complaints, when gridlocks have affected local arteries for quite a long time; and (ii) the workload from verification of incidents and

determination of signal plans: traffic operators need to gather and analyze incident information (e.g., location, lane closure types, etc.) from multiple directives, such as cameras and travel information platforms. In other words, incident plans are determined with considerable manual efforts of integration and analysis on the multi-source traffic data.

This paper proposes to improve incident management efficiency by introducing a decision-making framework that automates the data analysis process and learns to recommend signal plans even before official report of incidents. Specifically, instead of learning end-to-end mappings from road traffic states to action plans, the recommendation task is decomposed into two subtask

¹Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA

²Heinz College of Information Systems and Public Policy, Carnegie Mellon University, Pittsburgh, PA

Corresponding Author:

Sean Qian, seanqian@cmu.edu

models in hierarchy—traffic predictor and signal plan associator. For traffic prediction, real-time data inputs monitored from crowdsourced Waze alerts and traffic sensors are incorporated to trigger predictions of traffic delays in the network. A novel neural sequential learning model using encoder-decoder architecture with attention mechanism is developed for this task (2). For plan associator, to incorporate domain knowledge from developed incident timing plans, every incident plan is encoded into a matrix of plan keys characterized by its incident-triggering conditions, which are derived by transportation experts, and a normalizer converts traffic predictor outputs to queries. By defining various metric features for measuring the closeness between traffic query and plan keys, it is proposed to learn a linear kernel of metrics, which gives a higher ranking score for every historically engaged query key pair than the irrelevant ones. L_1 -regularized rank logistic regression model (RankLR) is used for this task. It is found that the recommendation system shows a precision score of 96.75% and recall of 87.5% on the testing plan, and an average of 22.5 min lead time ahead of Waze alerts for making plan recommendations. The results suggest that this decision-making framework is capable of giving traffic operators significant time to access the conditions and reacting appropriately.

Related Work

Traffic Signal Timings for Incident Management

Most TMCs respond to traffic incidents by placing variable message signs, closing lanes, or forcing turnings. Recently, studies have examined the optimization of traffic signal timings as an active management tool for non-recurring congestion (3–6). Traffic assignment models, equipped with behavior models that characterize travelers' behavior changes after incidents, and prediction-correction models are often built to simulate the time-dependent diverted traffic flow under pre-defined incident scenarios (7–13). The signal timings, optimized for a given incident scenario, then favor specified directional movements to minimize the induced congestion (14). This study, which is built upon developed incident timing plans, refines the decision-making process or transition logic of signal patterns, by recommending optimal incident signal plans to traffic operators ahead of time. The closest work to this is the work of Ban et al., which similarly considers real-time incidents, traffic volumes, and weather data for activation of signal control (15). However, they aim to determine if adaptive traffic control systems (ATCS) should be activated—while detailed signal timing plans were recommended to traffic operators. Their model learns to classify level-of-service (LOS) outcomes of signal control activation with equal amount

of before–after experiment data. Instead, the model is trained to replicate TMCs' manual operation, of which records are readily available even in small townships. Most importantly, they do not consider ahead-of-time recommendation with traffic prediction, which is one of the central contributions of this work.

Traffic Prediction

Data-driven models have become popular approaches for real-time traffic prediction. Recent models are built with spatiotemporal traffic flows, traffic events and incidents, and weather data to trigger traffic volume or speed prediction 5 to 30 min ahead (16–20). Historical average, linear models such as autoregression or LASSO, local regression and nearest neighborhood methods, graphical models, and deep neural networks are common modeling choices (16, 18–26). Encoder-decoder recurrent neural network is a popular deep learning architecture initially proposed for machine translation (27). Researchers have applied it for predicting traffic sequences (28, 29). Attention mechanisms are often embedded between encoder-decoder stacks to reduce the burden of compressing all observed information at each time step (2, 30). This study applies encoder-decoder Gated Recurrent Unit (GRU) with bilinear attention mechanism for predicting spatial traffic time-series in target traffic network (27, 30).

Metric Learning

Metric learning, whose goal is to find appropriate similarity measurements of data points, was initially proposed for recommendation system, such as search engines, to customize rankings with user clickthrough logs (31–33). It has been adapted to zero-shot learning to classify instances of unseen classes during training (34–39). Their approach is to project inputs and class attributes into the same feature space and associate them with a compatibility function with learned parameters. Metric learning is employed to tackle the cold-start problem of signal plan recommendation. The lack of expert records is expected to exist during initial enabling of timing plans, or when new plans are added for expansion of signalized intersections.

The Contribution of this Study

This study can be differentiated from prior work in three ways:

1. The incident response time of traffic operators is shortened by combining approaches from both traffic prediction and recommendation systems. Other work either predicts traffic without prescribing

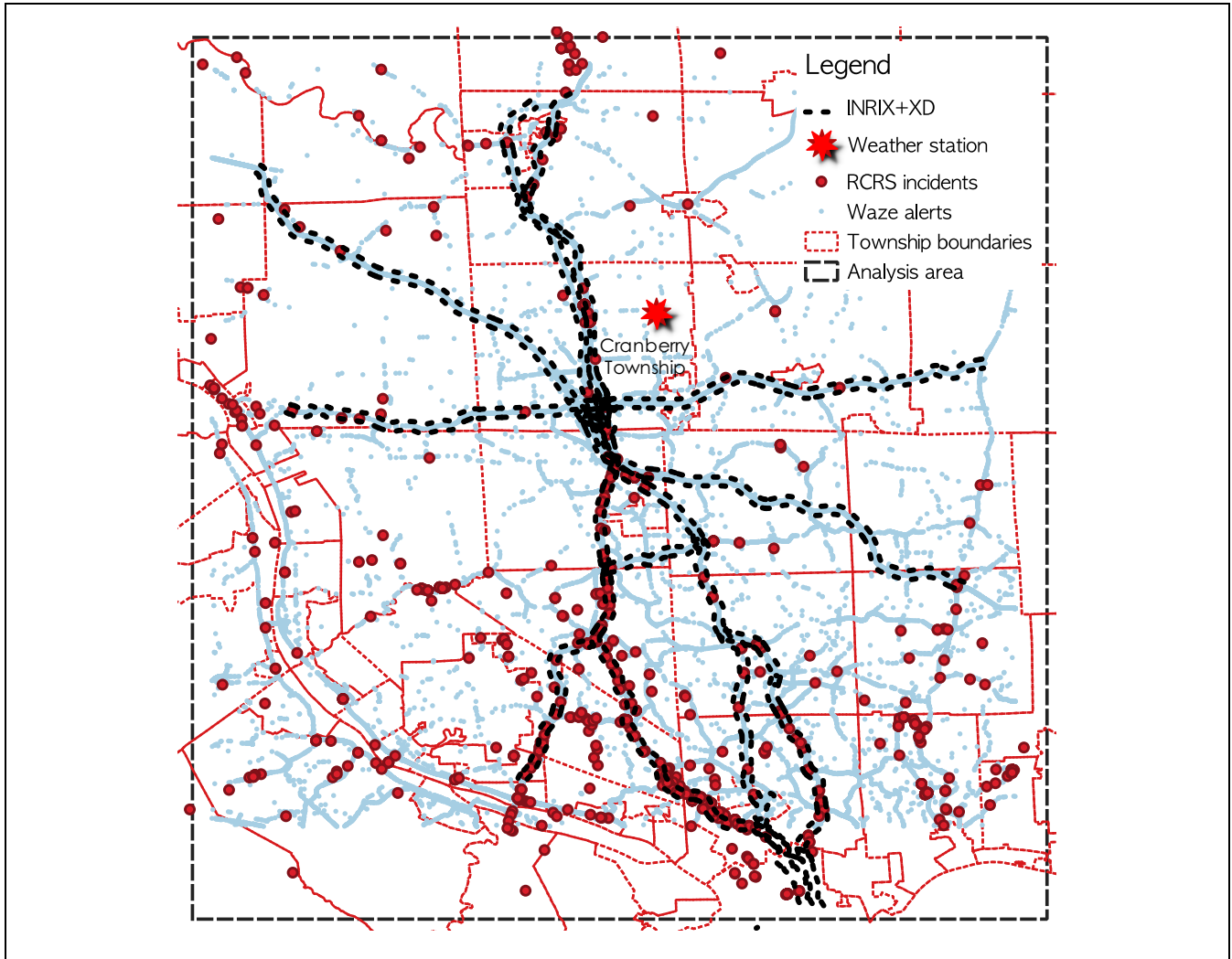


Figure 1. Data sources used in this paper.
 Note: RCRS = Road Condition Reporting System.

- actions, or determines signal timings with current road conditions.
2. A novel hierarchical model is proposed which learns to recommend incident signal timings to traffic operators with domain knowledge and very few historical demonstrations.
3. Our model processes crowdsourced data in real-time for traffic prediction and incident management. Few existing works present data processing and feature engineering methods for Waze data feeds.

Dataset and Preprocessing

This section describes the data sources used in this work, which includes INRIX probe traffic speed data (<http://inrix.com/products/ai-traffic/>), PennDOT Road Condition Reporting System (RCRS) incident report

(<https://www.penndot.gov/Doing-Business/OnlineServices/Pages/Developer-Resources.aspx>), Waze alerts (<https://www.waze.com/>), and Weather Underground (<https://www.wunderground.com/>). Figure 1 illustrates the data sources and collection area for Cranberry Township, Pennsylvania, U.S.

INRIX Traffic Speed

The INRIX traffic data were reported every 5 min for road segments georeferenced by INRIX XD code. Each data record includes the XD segment code, time stamp, observed speed (mph), average speed (mph), reference speed (mph), and two parameters for the confidence of the speed, namely confidence score and confidence value. INRIX data was downloaded between January 1, 2017 and July 21, 2019. All XD segments in Cranberry Township, and Interstate freeways XDs within 30 min

driving were selected in this study. The selected XD segments are shown in Figure 1. Missing values are imputed with the last speed observations of this segment.

Traffic Incident

Pennsylvania DOT (PennDOT) RCRS Incident Report. RCRS data feeds, shown as red dots in Figure 1, provide real-time information for traffic incidents, roadwork, winter road conditions, and other events which cover all state-owned roads in 511PA road network. An incident record includes the incident location, road closure types, causes, and close and open time stamps. RCRS reports between 2017 and 2019 were provided by PennDOT. The incident records without location coordinates, and those lasting for more than 24 h, are removed from the dataset.

Waze Alerts. Waze is a mobile navigation application that enables editing of the map with crowdsourced user reports. Users can report traffic crashes, congestion, hazards, or police traps on the road (40). Waze data (blue dots) from February 9, 2019 to July 23, 2019 were collected from Waze GeoRSS API. Only incidents reported under accident or jam categories were considered, to remove most of the false alarms. However, Waze is not screened regarding the reliability score to lower dependency on the external system.

Weather Underground

Weather Underground reports hourly weather measurements. Each entry contains temperature, pressure, dew point, humidity, wind speed, precipitation, pavement condition, and visibility, and so on. The position of the chosen weather sensor is shown in Figure 1. The weather time series are resampled every hour and missing values are imputed with linear interpolation.

Method

This section first describes the data processing steps. Then the model architecture is presented that learns to recommend incident timing plans by two decomposed learnable models, namely, traffic predictor and signal plan associator.

Data Processing

The proposed data processing pipeline integrates and transforms multi-source traffic speed, incident, weather, and temporal data into representative features for sub-task models. One-hot encoding for categorical variables is applied. All processed features are scaled by min–max normalization.

Speed Processing. Two segment-level features, travel time index (TTI_{itd}) and slowdown speed (SD_{itd}), are extracted from raw traffic speed data to describe road conditions. v_{itd} is used to denote the observed speed on XD i at time t on day d . To measure congestion on this segment, travel time index (TTI) is used, which is defined as real-time travel time divided by free-flow travel time, and can be computed by Equation 1. To determine the reference (free-flow) speed v_i^{ref} of an XD i , the 85 percentile of observed speed on that segment for all time periods (Equation 2) is used, which is the recommended approach for computing reference speed from probe-based speed data (41). A large value of TTI_{itd} indicates the segment is congested.

$$TTI_{it}^d = \max(v_i^{\text{ref}}/v_{itd}, 1) \quad (1)$$

$$v_i^{\text{ref}} = \mathbb{P}_{0.85}(v_{itd}) \quad (2)$$

To encode flow spillbacks in the network, slowdown speed (SD_{itd}) is proposed. Slowdown speed, defined in Equation 3, is computed by subtracting speed v_i from the mean speeds of N_i upstream XDs of i , denoted as $\Gamma^{-1}(i)$. A large value of SD_{itd} indicates that back-of-queue slowdowns exist on segment i and may infer the occurrence of traffic incident.

$$SD_{itd} = \max\left[\frac{\sum_{j \in \Gamma^{-1}(i)} v_{jtd}}{N_i} - v_{itd}, 0\right] \quad (3)$$

Incident Processing. An integration and processing method is proposed for multi-source incident data. It deserves notice that Waze differs from RCRS in that (i) Waze alerts are reported by road users, and often appear immediately after the occurrence of incidents, while RCRS documents TMCs' road closures in response to incidents, which are inputted after actions are taken; (ii) Waze contains geographical point features indicating the position of road users, while RCRS is usually line feature for the begin/end locations of road closure events; (iii) Waze has duplicate records for one incident, while RCRS is usually unique. Naturally, it is assumed that Waze and RCRS represent different status of a traffic incident: (i) road users first report incident occurrence on Waze; (ii) if the induced congestion calls for road closure, TMC then takes actions, such as placing barricades on the affected road, and documents it on RCRS; (iii) traffic goes back to normal (incident clear), and Waze and RCRS are removed from the feed.

The workflow in Figure 2 is thus developed to integrate multi-source incident information by their location and status. For incident location, incidents are first mapped to their affected road segments. A vector $\mathbf{x}_{td}^{\text{inc}} = [x_{1td}^{\text{inc}}, \dots, x_{ntd}^{\text{inc}}]$, where n is the number of segments in the network, is used to describe the spatial

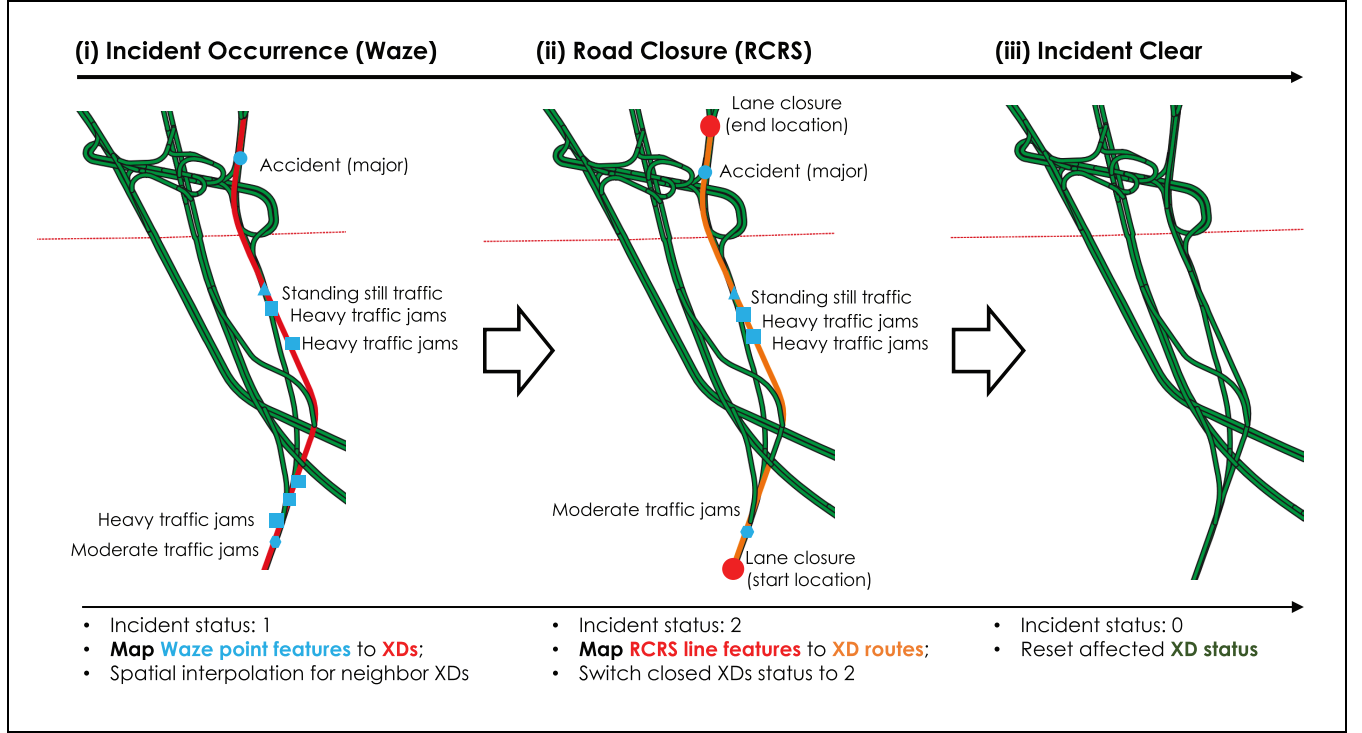


Figure 2. Workflow for integration of multi-source incidents.

Note: RCRS = Road Condition Reporting System.

incident location at t . Note that for processing Waze point features, an interpolation step is performed at each time step to fill spatial gaps between affected segments. The middle segment along the route is added to the incident-affected area if the shortest graph distance between both ends of two Waze alerts is just one neighbor. For incident status, each element $x_{itd}^{inc} \in \mathbf{x}_{td}^{inc}$ is encoded as a three-category variable, where 0 is normal traffic, 1 means the incident is reported by Waze, and 2 denotes the road closures reported by RCRS. A max-gate operator then integrates multi-source incidents status on segments at each time step, that is, the final value of x_{itd}^{inc} is the largest status value mapped to segment i by multi-source data. The proposed incident processing workflow is expected to represent a major incident as 1-2-0 or 1-2-1-0 along the time dimension, as shown in Figure 2.

Weather and Time Features. Weather features used in this paper include six continuous variables—temperature, humidity, wind speed, pressure, visibility, and hourly precipitation, and a binary variable—pavement condition. Time features include five categorical variables: time-of-day, week-of-year, month-of-year, day-of-week, and holiday. For the cyclic month, week-of-year and time-of-day categorical variables, sine and cosine functions are used to transform them into a two-dimension vector $[t_i^{(sin)}, t_i^{(cos)}]$:

$$t_i^{(sin)} = \sin(2\pi i/T) \quad (4)$$

$$t_i^{(cos)} = \cos(2\pi i/T) \quad (5)$$

where i denotes the week/month/time index and T denotes the total weeks/months/time steps in a cycle. An advantage of this “clockwise” encoding is that each variable is mapped onto a circle such that the lowest value for that variable appears right next to the largest value (e.g., January is right next to December). For day-of-week and holiday variables, one-hot encoding is applied after combining similar time features. Specifically, while Monday and Friday are encoded separately, Tuesday to Thursday are merged into one variable, so are Saturday, Sunday, and official holidays.

Model Architecture

The proposed incident plan recommender consists of two interconnected learnable models: traffic predictor and plan associator. The traffic predictor is an encoder-decoder recurrent neural network with attention mechanism, which takes the speed (and slowdowns), traffic incidents, weather, and time features and triggers predictions of traffic speed time-series on target segments for the future 30 min (27). To incorporate domain knowledge of the developed incident plans, the

signal timing manual is obtained from Cranberry Township Traffic Management Center and the plan triggering conditions are translated into a matrix of plan attributes (keys). The plan associator then generates traffic queries from current and 30-min predicted future speed series and their closeness with plan keys are evaluated with self-defined metrics. The plan associator learns to rank incident plans by fitting a linear kernel of the proposed metrics from historical engagement records. The module is named as “metric kernel” in the remainder of this paper.

Traffic Predictor. Predicting traffic beyond 5–10 min ahead is hard for traditional autoregressive time-series methods. Traffic on a road segment can change drastically because of traffic incidents, weather hazards, or atypical traffic patterns in its proximity. In these cases, past traffic dynamics on the target road segments may have little useful information implying their future traffic states. A widely used solution found in literature is to take into account spatiotemporal correlations between target road segments and nearby segments (19, 20, 42). As it takes time for traffic to propagate, abnormal traffic nearby can work as longer-term predictors. This paper models the traffic prediction problem as a sequence-to-sequence task, and the state-of-the-art architecture, encoder-decoder neural network with attention mechanism is built to predict the future traffic flow sequence up to 30 min, with a resolution of 5 min (2, 30).

As shown in Figure 3, the architecture is comprised of an encoder model, a decoder model, and an attention model which queries the encoder dynamically via a context vector. Both encoder and decoder models use GRU, which processes the feature input $\mathbf{X}_t$ sequentially into encoder hidden states $\mathbf{H}_t^{\text{enc}}$ and predicts the speed time-series $\hat{\mathbf{Y}}_{t+h}$ on all target segments (27). Denote h as the forecasting horizon, which ranges from 5 to 30 min. The definition of hidden state $\mathbf{H}_t^{\text{enc}}$ is consistent with Cho et al. (27). The encoder hidden states at the last time step $\mathbf{H}_t^{\text{enc}}$ is set as the initialized hidden states of decoder model. For decoding, the current speed on target segments $\mathbf{Y}_t$ is used as initial inputs to the model, and the predicted speed $\hat{\mathbf{Y}}_{t+1}$ are fed as inputs for next step predictions in an autoregressive way.

In the standard encoder-decoder models, the encoder model attempts to compress all the observed information at each time step into an intermediate representation of fixed size. One way to address this issue is via an attention mechanism, where references to the hidden states of the topmost encoder are kept and queried dynamically for decoding. There are also some physical intuitions for applying attention mechanism. For short-term traffic prediction, it is expected that traffic conditions on downstream segments close to the target segment are good

indicators. However, for longer-term prediction, traffic conditions on downstream segments of variable distances (usually far away from target segments), which depend on current and predicted network traffic conditions, should be dynamically attended to, as congestion takes time to propagate backwards. Therefore, a bi-linear attention mechanism proposed by Bahdanau et al. is used in this paper (30). At each time step, the decoder computes the attention weights for each encoder output $\mathbf{H}_t^{\text{enc}}$ by a bi-linear correlation kernel with learned weights. The context vector c_t in Figure 3 is the average of encoder output at each time step weighted by attention weights and is integrated with decoder hidden states to trigger the speed predictions.

Plan Associator. The plan associator learns to select from pre-defined decision-making rules, and combine them to recommend signal timing plans based on current and predicted traffic conditions on the target network. This paper proposes to learn similarity metric kernel between network traffic queries, which are transformed from current and predicted speed time-series on target segments, and keys, which are the triggering conditions of each incident plan. As shown in Figure 4, the plan associator architecture comprises of the encoding schemes of incident plan keys, the normalizer for network traffic prediction queries, and the learned metric kernel.

Incident Signal Timing Plan. The signal timing manual was obtained from TMC in Cranberry Township. Cranberry Township is located at the junction of Interstates 79 and 76 (PA Turnpike). The township owns 18 active and six proposed signalized intersections on local arterial roads US 19, Freedom Road (3020), and Route 228. Six coordinated signal timing plans for different incident scenarios, which involves the control of the 18 active signals, have been developed by the TMC and used in this study. These incident signal timing plans are prepared by (1) simulating the diverted traffic flow under pre-defined incidents using PTV Visum (<http://vision-traffic.ptvgroup.com/en-us/products/ptv-visum/>), and (2) optimizing signal timing function and coordination by Synchro Studio (<https://www.trafficware.com/synchro.html>), with the diverted travel demand as inputs. The developed plans are as follows:

- *Plan A* is an incident plan for managing incidents on I-79 southbound south of Exit 83 (Zelienople/Jackson Township SR 528) and north of Route 228. Traffic signal timings were developed to favor US 19 southbound, from Old Route 19/Victory Church Dr to Thorn Hill Rd. One entire network coordination zone with 175 s cycle lengths for both a.m. and p.m. peaks is activated;

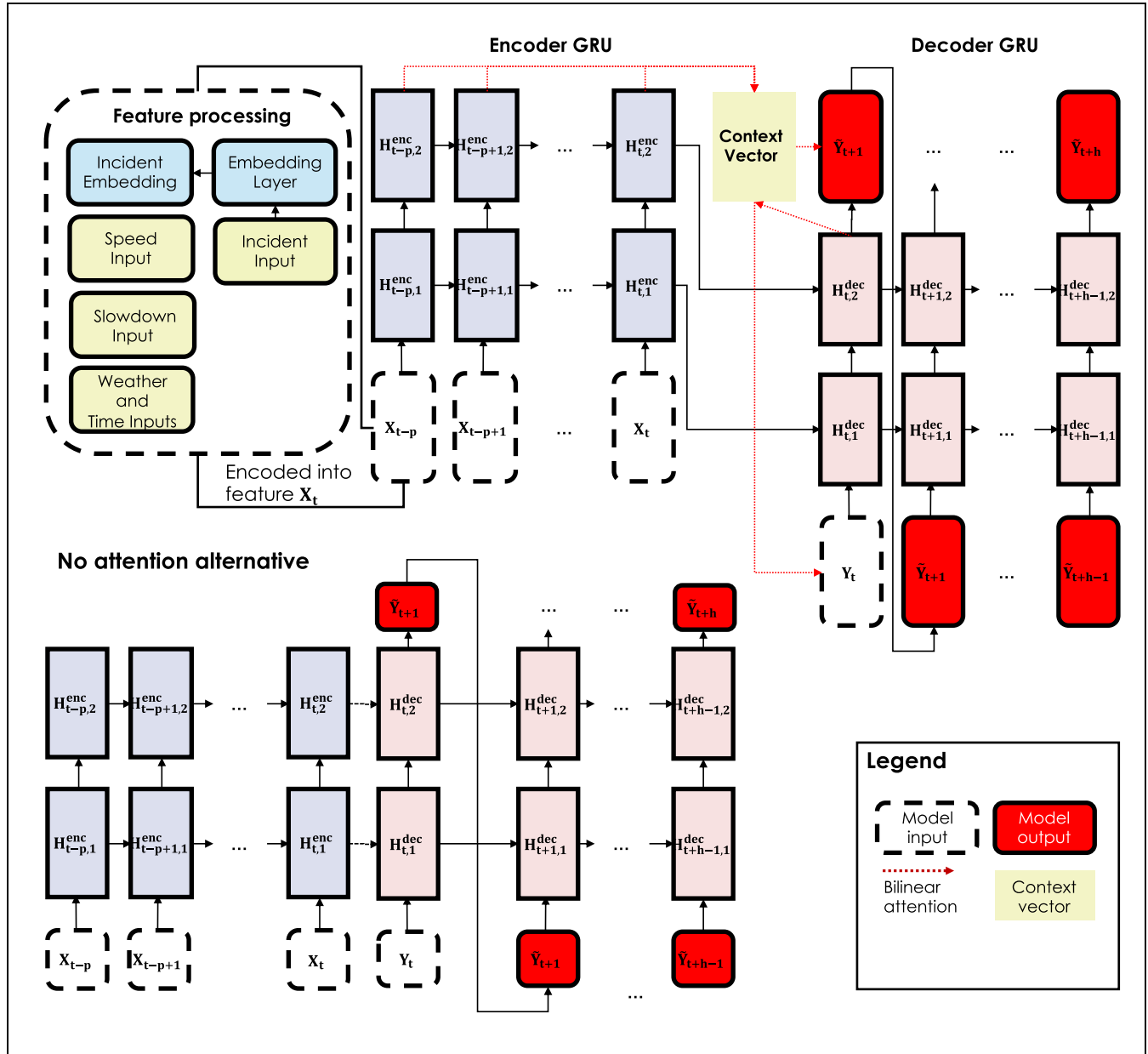


Figure 3. Traffic predictor architecture.

Note: GRU = gated recurrent unit.

- *Plan B* manages incidents on I-79 northbound and north of Route 228. One entire network coordination zone, with half-cycling where possible, is activated to favor US 19 northbound, from Emeryville Rd/Freeport Rd to Old Route 19/Victory Church Dr. a.m. Peak cycle length is 180 s (90 s) and p.m. Peak cycle length is 210 s (105 s);
- *Plan C* is for incidents on I-79 southbound and south of Route 228. One entire network coordination zone half-cycling is activated to favor US 19 southbound. Cycle lengths are 200 s (100 s) for both peaks;
- *Plan D* is for incidents on I-79 northbound and south of Route 228. One entire network coordination zone with 180 s cycle length is activated to favor movement on US 19 northbound;
- *Plan E* is for incidents on I-76 (PA Turnpike) eastbound and east of Cranberry Township. The majority of the traffic was assumed to be heading east on SR 228 towards SR 8 back towards I-76 (PA Turnpike). One entire network coordination zone with half-cycling (180/90 s) at the Turnpike Ramps and I-79 Ramps intersections is activated for p.m. peaks to favor this movement;

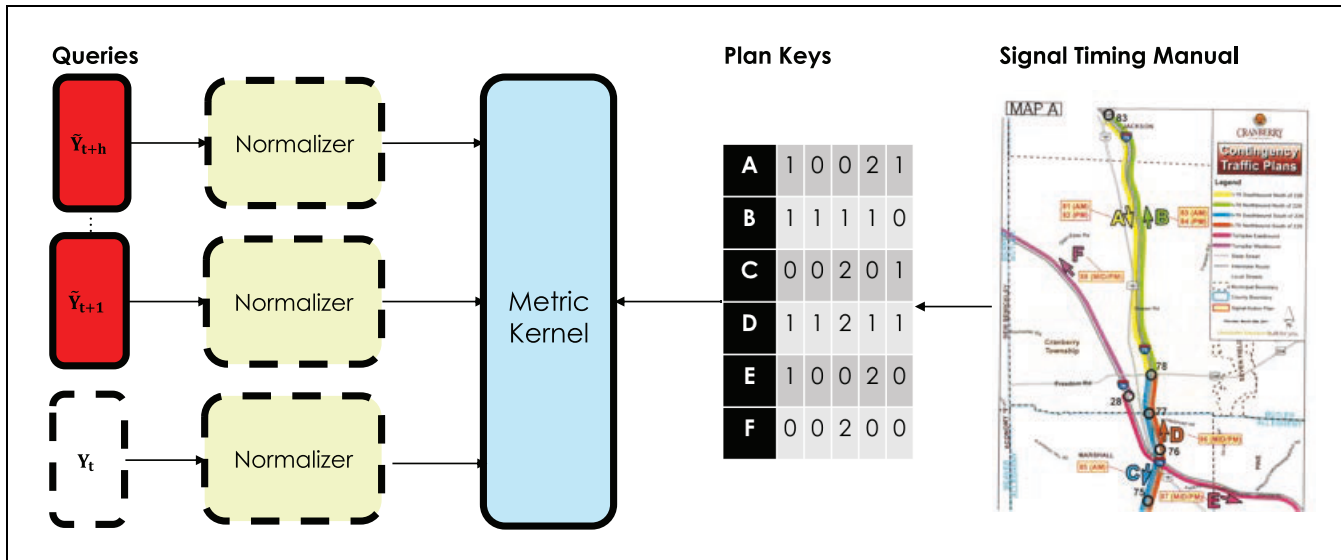


Figure 4. Plan associator architecture.

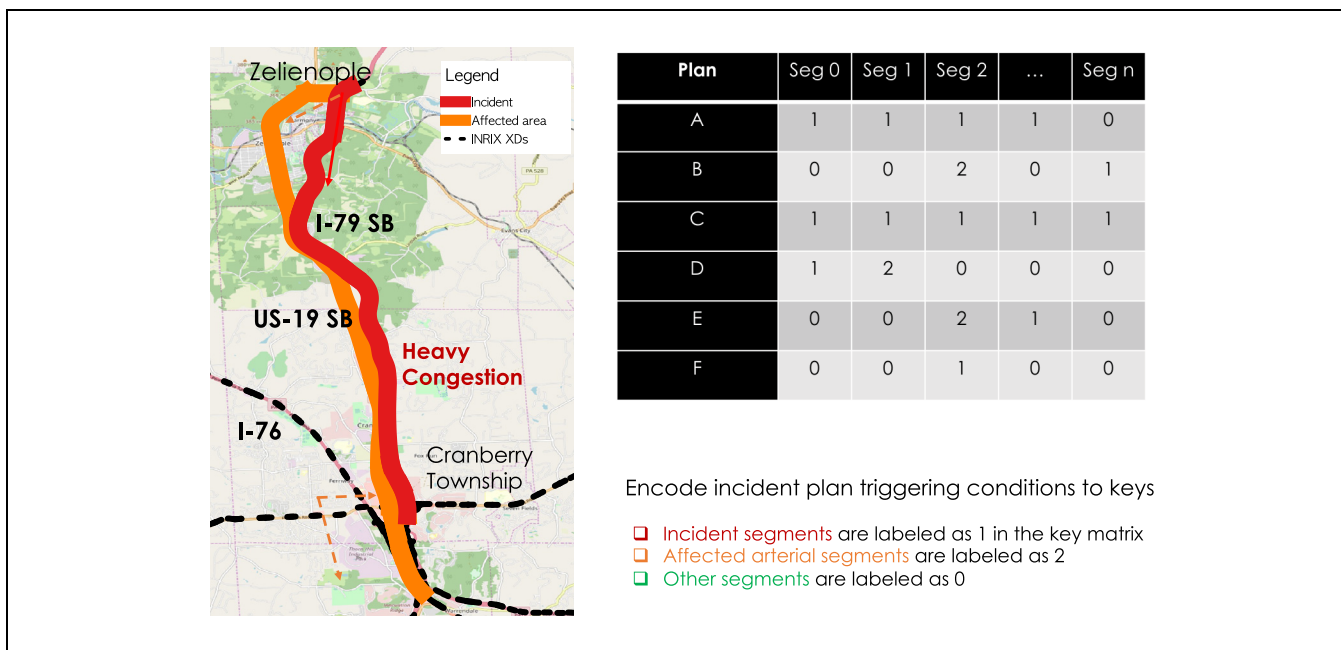


Figure 5. Encoding scheme of incident plan attributes and keys.

Note: SB = southbound.

- Plan F* is for incidents on I-76 westbound and west of Cranberry Township. The majority of the traffic was assumed to be heading west on Freedom Rd towards SR 65. One entire network coordination zone with half-cycling (180/90 s) at the Turnpike Ramps and I-79 Ramps intersections is activated to favor this movement during p.m. peaks.

Incident Plan Keys. The incident plan triggering conditions presented above were processed into plan key matrix. Traffic incidents (e.g., significant congestion or road closures) occurring on I-79 and I-76 are likely to spill back to the local network and causes catastrophic traffic gridlocks. For example, if a major incident occurs on I-79 SB north of Rt 228 causing severe congestion

(Plan A), as shown in Figure 5, traffic is expected to enter southbound US 19 at Zelenople and back up at Rt 228. A developed coordinated signal pattern A, which favors US 19 southbound movement, will be engaged. To incorporate the domain knowledge from developed incident plans, a plan key matrix $\mathbf{P}$ is built to characterize the plan attributes and triggering conditions. As illustrated in Figure 5, the entries of incident segment columns in $\mathbf{P}$ is set as 1 and affected arterial segment columns set as 2. A null plan is added to the matrix, with all columns set to 0, to represent that no plan is to implement.

Normalizer for Network Traffic Queries. The speed outputs of traffic predictor are transformed to TTI by Equation 1. This intermediate step transforms traffic predictor outputs into a same scale for every segment.

Metric Learning. To learn a metric kernel that associates network traffic queries to plan key matrix, several metrics are defined beforehand to characterize the closeness between traffic queries and plan keys, and a linear function of these metrics is fit, which reinforces partial-order preference observed in engagement records. The metric kernel learns to select from pre-defined metrics, or decision rules, and combine them to determine signal timing plans. Three groups of metrics are defined to evaluate the relevance between network traffic queries at each time step and plan keys, which include:

1. *Triggering precision:* A threshold set $\{\text{TTI}^{\text{thres}}\}$ is pre-defined to detect if the query items exceed the threshold. The incident key matrix is also binarized to 0–1. For incident plans A–F, if any item of incident segments in queries is 1, the metric outputs 1 and otherwise 0. For null plan, the precision score between binarized key matrix and thresholded query vector are evaluated (0 is positive label), because the triggering segments of null plan are the whole target network. TTI thresholds, including 1.6, 2, 2.5, 5, and 10 are used.
2. *Rule:* After thresholding and binarizing the query, **if** the overlapped terms between query and binarized key vector contain both incident and affected arterial segments, **then** their relevance is 1 and **otherwise** 0. The same set of thresholds are used.
3. *Similarity:* The query vector is first upper-clipped by the threshold and scaled to 0–1 by min-max normalization. The inverse of euclidian distance between binarized key matrix and normalized query matrix is evaluated. The same set of thresholds are used.

The derived 15 metrics between a network traffic query q_k and plan key p_k are evaluated for 7 (6 predicted

and 1 current) time stamps in traffic queries and concatenated into $\mathbf{x}(q_k, p_i) \in \mathbb{R}^{105}$. RankLR proposed in Bahdanau et al. is applied to fit a linear kernel of the developed metrics (33). A dataset D is built containing all network traffic query and plan pairs in the records, with the engaged pairs in $D_+^{(k)}$ and irrelevant ones in $D_-^{(k)}$. A pairwise dataset is created by drawing any pair i from $D_+^{(k)}$ and one j from $D_-^{(k)}$, evaluating their respective metric vector $\mathbf{x}(q_k, p_i)$ and $\mathbf{x}(q_k, p_j)$, and computing the difference $\mathbf{x}_{ij}^p = \mathbf{x}(q_k, p_i) - \mathbf{x}(q_k, p_j)$. These $\mathbf{x}_{ij}^p$ are set as positive samples for pairwise learning. A reverse operation is also conducted by choosing one pair from $D_-^{(k)}$ and one from $D_+^{(k)}$, and these are set as negative samples. L_1 -regularized logistic regression is applied on the pairwise dataset to fit a linear kernel of the developed metrics so that the log loss in Equation 6 is minimized. The algorithm finds $\mathbf{w}$ that gives a higher ranking score for every relevant query–key pair than the irrelevant ones.

$$\min_{\mathbf{w}} \frac{1}{P} \sum_{i=1}^P [y^p \log(\mathbf{w}^T \mathbf{x}_{ij}^p) + (1 - y^p) \log(1 - \mathbf{w}^T \mathbf{x}_{ij}^p)] + C \|\mathbf{w}\|_1 \quad (6)$$

Determination of Incident Signal Plan. Recommendation score s_{ij} of a query–key pair is defined in Equation 7 as the linear combination of the defined metric evaluations, weighted by the learned $\mathbf{w}$. The incident signal plan with the highest recommendation score is activated or transitioned to. For transition between signal timing plans, a trigger of at least 20 min has been added between pattern changes as the system takes time to react, and traffic control can be significantly less efficient during the transition. Note that the incident signal timings are turned off if null plan is activated.

$$s_{ij} = \mathbf{w}^T \times (q_k, p_j) \quad (7)$$

Hyperparameters and Training. The encoder-decoder neural network is implemented in PyTorch. The following are used for all GRUs: Tanh activation, 256 hidden dimensions, 2 layers, and recurrent dropout of 0.2. The embedding layer for incidents has 3 dimensions. Attention has 256 hidden states. The network is trained using Adam optimizer for a maximum of 200 epochs and early-stops if validation error does not decrease for 5 epochs. Learning rate of 0.0005, teacher-forcing ratio of 0.5 and mini-batch size of 32 are used. For RankLR, an L_1 penalty of $C = 1$ is used for all models.

Baselines. Four traffic prediction baselines are experimented with.

Historical-Average. A baseline which uses day-of-week speed profiles averaged over the past 1-month window as speed predictions for the future 30 min.

Latest-Observation. A baseline which uses the latest observed speed on the segment as speed predictions for the future 30 min.

LASSO. Linear regression models with L_1 regularization, that is, LASSO, which use the same feature set as the neural network model, are built for each segment i and prediction horizon h independently (19, 43). The model learns the weights $w_h^{(i)}$ such that the loss in Equation 8 is minimized. L_1 regularization helps the model select the most critical features that are linearly related to the response. L_1 penalty hyperparameter α_{ih} controls the number of selected features, and is tuned by cross validation on the training set.

$$\min_{w_h^{(i)}} \|y_{t+h}^{(i)} - [X_{t-p}, \dots, X_t]w_h^{(i)}\|_2^2 + \alpha_h^{(i)} \|w_h^{(i)}\|_1 \quad (8)$$

GRU-No-Attention. A non-linear GRU model baseline. This baseline removes the attention mechanism in the model. The same hyper-parameters of the model are used.

Experiments

The experiments were run on a Linux workstation with a P4000 GPU. Traffic and multi-source data from 5:30 a.m. to 8:55 p.m. are selected, which have in total 29,520 data samples. Four signal engagement demonstrations, one for each incident plan (A,C,D,F), are available for training and evaluation. It deserves notice that the whole four days' data, during which incident signal timings were engaged, was removed for training the traffic predictor. The experiment setting ensures that ground-truth traffic conditions are not abused for prescribing incident plans. However, drawbacks are that traffic predictor only learns traffic dynamics when no human impact, that is, manual changes of signal plans, is involved. In this study, it is assumed that delayed actions of traffic operators do not impact traffic dynamics much during catastrophic traffic gridlocks. Therefore, the proposed models are suitable in initial enabling phase of the decision-making framework, when late response of traffic operators still exists. However, when traffic operator's response time is largely reduced, new traffic predictor considering manual change of signal plans should be built.

For testing traffic prediction and incident plan recommendation performances, different evaluation methods are adopted:

- For traffic predictor, 80% of the data samples are used for training and 20% for testing. Hyper-parameters of LASSO are tuned by 5-fold cross-validation on the training samples. Other baselines do not require tuning. root-mean-square error (RMSE) and mean absolute percentage error (MAPE) of speed predictions on the test set is computed for model comparison. Note that when computing MAPE, the absolute error is deliberately divided by the predicted values, so that negative errors (e.g., unable to predict congestion growth) are with a heavier penalty. The MAPE is defined in Equation 9:

$$\text{MAPE}_t = \frac{1}{n} \sum_{i=1}^n \frac{|A_t - F_t|}{F_t} \quad (9)$$

where A_t is the actual value and F_t is the forecast value.

- For the recommender model, first it is fit with all engagement records to examine the model weights. Then, leave-one-out evaluation is applied, that is, every three engagement records are used for training plan associator and the recommendation is evaluated on the remaining record. Macro precision is computed and scores of the recommended plans for all time stamps during signal engagement periods are recalled, and the whole recommender scoring behaviors are visualized for each testing plan.

Traffic Prediction

Results of this model against other methods are presented in Tables 1 and 2 and visualized by segment in Figure 6. It is found that the encoder-decoder-attention model outperforms speed prediction baselines for all prediction horizons within 30 min, and the performance improvement is more obvious for longer-term prediction and for predicting growth of traffic. When comparing the RMSE, as shown in Table 1, for 5-min ahead prediction, very less usage of incorporating multi-source data and applying complex model architecture is found, since *latest-observation* method is almost of the same performance as the best model. However, for longer-term prediction, such as those larger than 10 min, *latest-observation* performs even worse than *historical-average* which does not use any real-time data. *GRU-no-attention* shows similar performances to *LASSO* on this dataset, but the model with attention mechanism added presents much better results for longer-term predictions. However, when compared with MAPE in Table 2, the proposed model outperforms other methods significantly. The results show that the model gains performance improvement

Table 1. RMSE Prediction Error (mph) of the Encoder-Decoder-Attention Model against Other Baselines for Different Prediction Horizons on the Test Samples

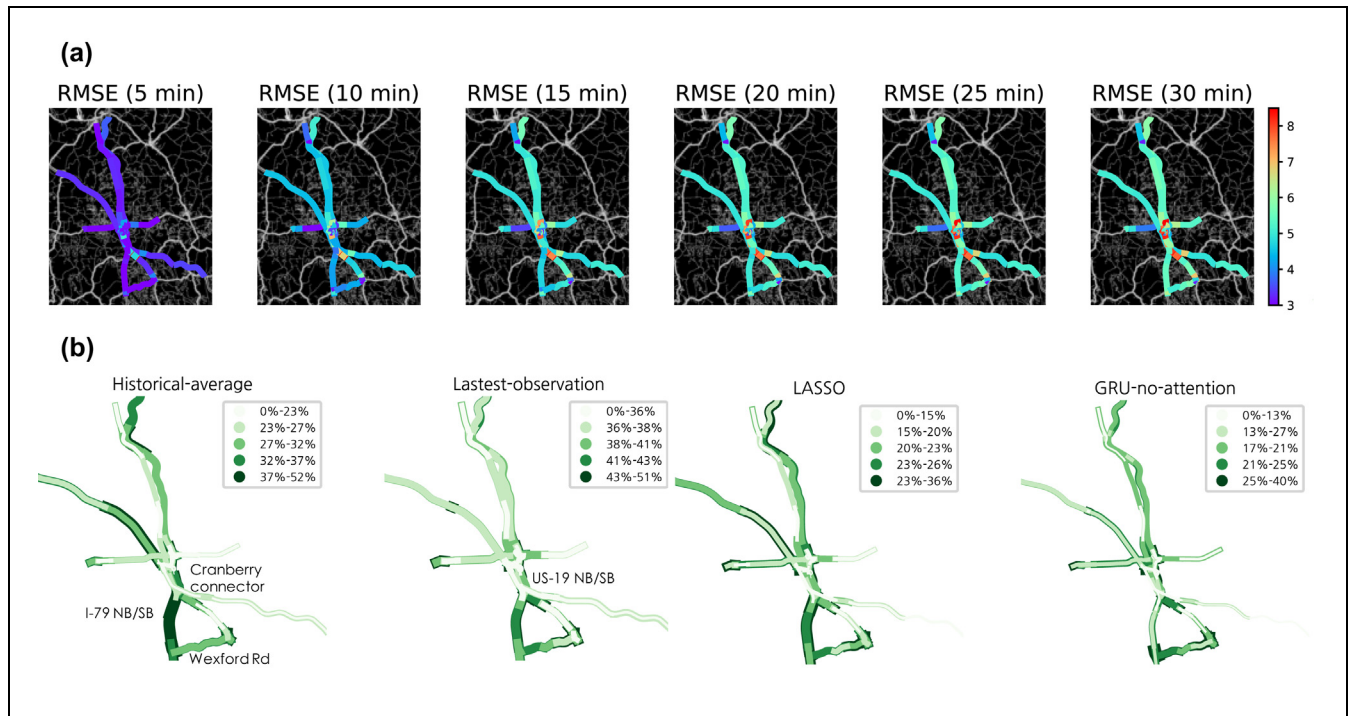
Model	5 min	10 min	15 min	20 min	25 min	30 min
Encoder-decoder-attention (our model)	3.187	3.579	3.461	3.388	3.417	3.623
Historical-average	5.306	5.306	5.306	5.306	5.306	5.306
Latest-observation	3.546	5.008	5.609	5.848	5.987	6.092
LASSO	3.297	4.207	4.460	4.557	4.606	4.647
GRU-no-attention	3.343	4.129	4.318	4.404	4.470	4.541

Note: Bolded values indicate the best performance among all models. RMSE = root mean square error; GRU = gated recurrent unit.

Table 2. MAPE Prediction Error of the Encoder-Decoder-Attention Model against Other Baselines for Different Prediction Horizons on the Test Samples

Model	5 min	10 min	15 min	20 min	25 min	30 min
Encoder-decoder-attention (our model)	8.39%	8.43%	9.24%	9.48%	9.66%	9.78%
Historical-average	11.73%	11.73%	11.73%	11.73%	11.73%	11.73%
Latest-observation	9.05%	15.80%	18.92%	20.35%	21.22%	21.94%
LASSO	8.05%	13.43%	15.63%	16.02%	17.46%	17.52%
GRU-no-attention	8.13%	11.05%	11.48%	12.30%	13.09%	12.05%

Note: Bolded values indicate the best performance among all models. MAPE = mean absolute percentage error; GRU = gated recurrent unit.

**Figure 6.** Visualization of traffic speed prediction model performances: (a) speed prediction performance of *encoder-decoder-attention* with 5–30 min prediction horizon, and (b) percentage improvement of the model against other baselines for 30-min ahead speed prediction.

Note: RMSE = root mean square error; GRU = gated recurrent unit; NB = northbound; SB = southbound.

mostly from reducing negative errors, that is, better at predicting congestion growth.

To further locate the sources of performance improvement, the percentage RMSE improvement of the

Table 3. Plan Associator Kernel Weights (Feature Name: Horizon-Thres-Metric)

Feature	Weight	Feature	Weight	Feature	Weight
5 min-2-similarity	2.098	0 min-1.6-rule	0.321	0 min-2.5-rule	0.073
10 min-2-similarity	2.07	5 min-2.5-rule	0.318	20 min-2-rule	0.071
5 min-2-precision	1.459	15 min-5-rule	0.293	30 min-2-similarity	0.059
25 min-1.6-rule	1.318	0 min-2-rule	0.29	25 min-1.6-precision	0.052
10 min-5-similarity	1.062	10 min-1.6-rule	0.239	0 min-10-similarity	0.044
25 min-10-similarity	0.746	5 min-1.6-precision	0.202	15 min-2.5-rule	0.04
20 min-10-similarity	0.593	15 min-2-rule	0.197	20 min-2-precision	0.038
25 min-2.5-precision	0.577	20 min-5-rule	0.177	15 min-1.6-precision	0.037
15 min-2-precision	0.539	5 min-1.6-rule	0.17	20 min-2.5-similarity	0.036
20 min-1.6-rule	0.535	30 min-1.6-rule	0.138	25 min-2-rule	0.033
30 min-2.5-rule	0.425	5 min-10-similarity	0.124	25 min-2.5-rule	0.029
25 min-5-rule	0.421	20 min-2.5-precision	0.118	30 min-1.6-precision	0.028
30 min-5-precision	0.403	25 min-5-similarity	0.092	30 min-2.5-precision	0.015
30 min-2-rule	0.375	5 min-5-similarity	0.083	0 min-1.6-precision	0.015
15 min-1.6-rule	0.36	10 min-2.5-rule	0.081	10 min-2-rule	0.012
10 min-10-similarity	0.339	15 min-2.5-precision	0.078		
0 min-5-rule	0.338	30 min-10-distance	0.078		

proposed method is visualized against other baselines by road segment in Figure 6. It is found that most improvement comes from better predictions of traffic on I-79 NB/SB, Wexford Rd and US-19 NB, and Cranberry Connector. I-79 SB/NB south of Rt 228 are two main sources of traffic incidents in Cranberry Township. A common incident-induced congestion pattern in Cranberry Township is that an incident on I-79 SB/NB closes part of the road, and traffic flows into US-19 as an alternative, therefore routing to/from it via Wexford Rd. The proposed model learns to capture this pattern better than other methods. Note that this property is very useful for recommending proper incident plan ahead of time. However, if compared with *GRU-no-attention*, most improvement comes from better predictions on Cranberry Connector Ramp. This might suggest that *GRU-no-attention* captures the easier I-79 incident-induced traffic pattern well, but is hard to compress the traffic flow states in the complex Cranberry Connector without attention.

Incident Plan Recommendation

The metric kernel is fit using all engagement records to examine the power of pre-defined metrics. The signal engagement periods, in total, span across 85 time stamps (425 min) and 41,334 (85×425) combinations of query-key evaluation pairs can thus be generated. The model selects 48 features from the 105 pre-defined evaluation metrics. Clearly, as shown in Table 3, the selected features are reasonable. They are across all three types of metrics, different TTI thresholds, and prediction horizons, as these features together are expected to stably determine the occurrences and clearance of congestion on plan triggering segments. In addition, all of the

selected features have positive weights, indicating that the selected metrics alone are effective for measuring the association between network traffic queries and incident plan keys. Interestingly, it is found that instead of using current observations, most powerful metrics with larger weights are evaluated on the predicted traffic series across multiple horizons. This also meets expectations, as a trend of traffic conditions are usually easier for decision making.

The generalization performances of plan recommender is then evaluated using leave-one-out strategy. Since each incident plan has only one engagement record, by using this evaluation method, the training and testing set becomes disjointed so that unseen class labels during training appear in the testing set. This issue is known as the zero-shot learning problem. Although end-to-end classification methods cannot generalize to unseen classes, the proposed method is expected to do it by learning an intermediate association between class attributes (incident plan keys) and features (network traffic queries) instead. In other words, the metric kernel learns to project network traffic queries into the space of plan keys. As shown in Figure 7, the recommender triggers appropriate plans for all of the four test cases, and also for switching between null plan and incident plans. The recommendation precision and recall during signal engagement periods, if averaged over four test cases and all time stamps, are 96.75% and 87.5%. The inconsistency between the stopping period of engagement causes the relatively low recall, where the recommender often stops the plan earlier than traffic operators. In addition, it is also found that, if compared with the earliest incident report time from Waze in Figure 7, the proposed model can trigger recommendations 10 min ahead for

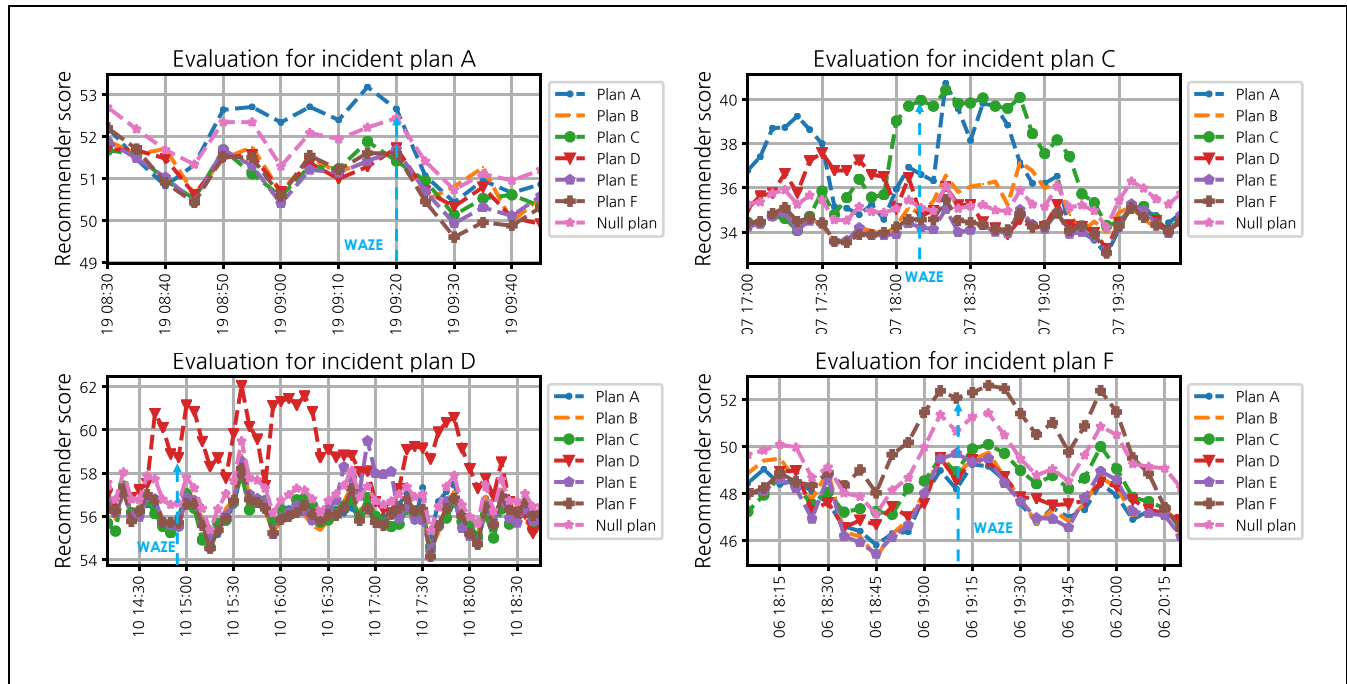


Figure 7. Visualization of plan recommender model performances.

plan C, 35 min ahead for plan F, 30 min for plan A, and 15 min for plan D. An average of 22.5 min advance recommendation performance can be achieved by the proposed method.

Conclusion

This paper proposes a novel decision-making framework which incorporates real-time data inputs monitored from crowdsourced Waze alerts and traffic sensors for traffic prediction, and constructs a learnable recommendation system for triggering incident signal plans ahead of time with the outputs from traffic predictor. The novelty of this work comes from decomposition of the end-to-end incident plan classification problem into two subtask models, that is, a traffic predictor that outputs network traffic time-series for the future 30-min horizon, and a plan associator, which transforms incident plan triggering conditions and predicted network traffic series into keys and queries with domain knowledge, and learns to associate them from historical signal plan engagement records.

The traffic prediction task is formulated as a sequential learning problem in this paper, and lagged spatio-temporal traffic speed, traffic incidents, weather, and time features are embedded as source inputs for predicting traffic speed series on target segments. A new data processing pipeline is developed for incorporating multi-source incident feeds from crowdsourced Waze alerts and PennDOT incident reports by their location and

status, and they are encoded into representative data features. A GRU neural network model using encoder-decoder architecture with bilinear attention mechanism is proposed for this task. Results show that the proposed traffic prediction model outperforms other baselines, especially for longer-term traffic prediction and for predicting incident-induced congestion. The sources of prediction improvement are tracked and the model is found to be capable of capturing typical traffic patterns on the dataset. Domain knowledge from developed incident timing plans is incorporated to constrain the model learning. An encoding scheme for transforming triggering conditions of each incident plan into the plan attribute keys is proposed, and a normalizer for converting traffic predictor output to traffic queries is applied. By defining various metric features for measuring the closeness between traffic query and plans, it is proposed to learn a linear kernel of the proposed metrics, which gives a higher ranking score for every relevant query-key pair than the irrelevant ones from engagement records. RankLR model with L_1 penalty is used for this task. It is found that the selected metric features are reasonable and are across all metric types, travel time index thresholds, and prediction horizons. The recommendation system shows a precision score of 96.75% and recall of 87.5% on unseen testing plans. This model can trigger an average of 22.5 min advance recommendation, in particular, 10 min ahead for plan C, 35 min ahead for plan F, 30 min for plan A, and 15 min for plan D. The proposed framework is expected to give traffic operators significant time to access the conditions and react

appropriately. In addition, the recommender has been shown to effectively recommend unseen plans in training. This generalization property makes this method an appropriate initializer for cold-start recommendation of new incident plans without engagement records, which may be created recently for expansion of signalized intersections.

Acknowledgments

The authors specially thank Jason Dailey, Kelly Maurer, and Marty McKinney from Cranberry Township Public Works for providing the contingency signal plan document and engagement records.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: W. Yao, S. Qian; data collection: W. Yao; analysis and interpretation of results: W. Yao, S. Qian; draft manuscript preparation: W. Yao, S. Qian. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is funded by Carnegie Mellon University's Traffic21 Institute and Mobility21. Mobility21 is a national University Transportation Center on mobility funded by U.S. Department of Transportation.

References

1. FHWA. *Reducing Non-Recurring Congestion*, 2019. https://ops.fhwa.dot.gov/program_areas/reduce-non-cong.htm. Accessed June 7, 2019.
2. Luong, M. T., H. Pham, and C. D. Manning. Effective Approaches to Attention-Based Neural Machine Translation. *arXiv preprint arXiv:150804025*, 2015.
3. Mao, T., A. S. Mihaita, and C. Cai. Traffic Signal Control Optimization under Severe Incident Conditions Using Genetic Algorithm. *arXiv preprint arXiv:190605356*, 2019.
4. Abudayyeh, D., D. Ngoduy, and A. Nicholson. Traffic Signal Optimisation in Disrupted Networks with Re-Routing. *Transportation Research Procedia*, Vol. 34, 2018, pp. 195–202.
5. Gordon, R. Non-Recurrent Congestion: Improvement of Time to Clear Incidents. In *Intelligent Transportation Systems*, Springer, pp. 41–90.
6. Blandin, S., and L. Wynter. Method and System for Optimizing Road Traffic Control in the Presence of Incidents. US Patent 10,297,149, 2019.
7. Qian, Z. S., W. Shen, and H. Zhang. System-Optimal Dynamic Traffic Assignment with and without Queue Spillback: Its Path-Based Formulation and Solution via Approximate Path Marginal Cost. *Transportation Research Part B: Methodological*, Vol. 46, No. 7, 2012, pp. 874–893.
8. Pi, X., and Z. S. Qian. A Stochastic Optimal Control Approach for Real-Time Traffic Routing Considering Demand Uncertainties and Travelers' Choice Heterogeneity. *Transportation Research Part B: Methodological*, Vol. 104, 2017, pp. 710–732.
9. Ma, W., and Z. S. Qian. On the Variance of Recurrent Traffic Flow for Statistical Traffic Assignment. *Transportation Research Part C: Emerging Technologies*, Vol. 81, 2017, pp. 57–82.
10. Qian, Z. S., and H. M. Zhang. A Hybrid Route Choice Model for Dynamic Traffic Assignment. *Networks and Spatial Economics*, Vol. 13, No. 2, 2013, pp. 183–203.
11. Zhu, S., D. Levinson, H. X. Liu, and K. Harder. The Traffic and Behavioral Effects of the I-35W Mississippi River Bridge Collapse. *Transportation Research Part A: Policy and Practice*, Vol. 44, No. 10, 2010, pp. 771–784.
12. Danczyk, A., X. Di, H. X. Liu, and D. M. Levinson. Unexpected versus Expected Network Disruption: Effects on Travel Behavior. *Transport Policy*, Vol. 57, 2017, pp. 68–78.
13. He, X., and H. X. Liu. Modeling the Day-to-Day Traffic Evolution Process after an Unexpected Network Disruption. *Transportation Research Part B: Methodological*, Vol. 46, No. 1, 2012, pp. 50–71.
14. Koonce, P., and L. Rodegerdts. *Traffic Signal Timing Manual*. Technical Report. Federal Highway Administration, Washington, D.C., 2008.
15. Ban, X., J. Wojtowicz, and W. Li. *Decision-Making Tool for Applying Adaptive Traffic Control Systems*. Technical Report. New York State Energy Research and Development Authority, 2016.
16. Yao, W., and Z. S. Qian. Real-Time Traffic Monitoring and Prediction for Cranberry Township. National University Transportation Center for Improving Mobility (Mobility21), February 12, 2019. <https://trid.trb.org/view/1583682>.
17. Yao, W., Y. Wang, N. Wang, and G. Yang. Prediction of Benefits of Special Taxi-Pooling Design for Large Transport Terminals: Case Study of Beijing West Railway Station. *Transportation Research Record Journal of the Transportation Research Board*, 2016. 2542: 33–44.
18. Qian, Z. S., P. Zhang, and W. Yao. User-Centric Interdependent Urban Systems: Using Energy Use Data and Social Media Data to Improve Mobility. Technologies for Safe and Efficient Transportation University Transportation Center (T-SET), October, 2018. <https://trid.trb.org/view/1564454>.
19. Yang, S., and S. Qian. Understanding and Predicting Travel Time with Spatio-Temporal Features of Network Traffic Flow, Weather and Incidents. *IEEE Intelligent Transportation Systems Magazine*, Vol. 11, No. 3, 2019, pp. 12–28.
20. Min, W., and L. Wynter. Real-Time Road Traffic Prediction with Spatio-Temporal Correlations. *Transportation Research Part C: Emerging Technologies*, Vol. 19, No. 4, 2011, pp. 606–616.

21. Smith, B., and M. Demetsky. Multiple-Interval Freeway Traffic Flow Forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, 1996. 1554: 136–141.
22. Zhang, P., and Z. S. Qian. User-Centric Interdependent Urban Systems: Using Time-of-Day Electricity Usage Data to Predict Morning Roadway Congestion. *Transportation Research Part C: Emerging Technologies*, Vol. 92, 2018, pp. 392–411.
23. Zheng, Z., D. Wang, J. Pei, Y. Yuan, C. Fan, and F. Xiao. Urban Traffic Prediction Through the Second Use of Inexpensive Big Data from Buildings. *Proc., 25th ACM International on Conference on Information and Knowledge Management*, ACM, 2016, pp. 1363–1372.
24. Sun, S., C. Zhang, and G. Yu. A Bayesian Network Approach to Traffic Flow Forecasting. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 1, 2006, pp. 124–132.
25. Lv, Y., Y. Duan, W. Kang, Z. Li, and F.-Y. Wang. Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 2, 2014, pp. 865–873.
26. Ma, X., Z. Dai, Z. He, and J. Ma. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors*, Vol. 17, No. 4, 2017, p. 818.
27. Cho, K., B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*, 2014.
28. Ma, Y., X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha. Trafficpredict: Trajectory Prediction for Heterogeneous Traffic-Agents. *Proc., AAAI Conference on Artificial Intelligence*, Vol. 33, 2019, pp. 6120–6127.
29. Yang, S., W. Ma, X. Pi, and S. Qian. A Deep Learning Approach to Real-Time Parking Occupancy Prediction in Transportation Networks Incorporating Multiple Spatio-Temporal Data Sources. *Transportation Research Part C: Emerging Technologies*, Vol. 107, 2019, pp. 248–265.
30. Bahdanau, D., K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473*, 2014.
31. Chechik, G., V. Sharma, U. Shalit, and S. Bengio. Large Scale Online Learning of Image Similarity through Ranking. *Journal of Machine Learning Research*, Vol. 11, 2010, pp. 1109–1135.
32. Joachims, T. Optimizing Search Engines Using Click-through Data. *Proc., 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, pp. 133–142.
33. Sculley, D. Large Scale Learning to Rank. *Proc., NIPS 2009 Workshop on Advances in Ranking*, 2009, pp. 1–6.
34. Akata, Z., F. Perronnin, Z. Harchaoui, and C. Schmid. Label-Embedding for Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 38, No. 7, 2015, pp. 1425–1438.
35. Akata, Z., S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of Output Embeddings for Fine-Grained Image Classification. *Proc., IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, New York, pp. 2927–2936.
36. Frome, A., G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A Deep Visual-Semantic Embedding Model. In *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), Curran Associates, Inc., 2013, pp. 2121–2129.
37. Romera-Paredes, B., and P. Torr. An Embarrassingly Simple Approach to Zero-Shot Learning. *Proc., International Conference on Machine Learning*, 2015, pp. 2152–2161.
38. Xian, Y., Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent Embeddings for Zero-Shot Classification. *Proc., IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 69–77.
39. Socher, R., M. Ganjoo, C. D. Manning, and A. Y. Ng. Zero-Shot Learning Through Cross-Modal Transfer. *Proc., Advances in Neural Information Processing Systems*, 2013, pp. 935–943.
40. Amin-Naseri, M., P. Chakraborty, A. Sharma, S. B. Gilbert, and M. Hong. Evaluating the Reliability, Coverage, and Added Value of Crowdsourced Traffic Incident Reports from Waze. *Transportation Research Record: Journal of the Transportation Research Board*, 2018. 2672: 34–43.
41. Jha, K. Determining Reference Speed from Probe-Based Travel Speed Data. Master's thesis, Texas A & M University, 2017. <http://hdl.handle.net/1969.1/161455>.
42. Ermagun, A., and D. Levinson. Spatiotemporal Traffic Forecasting: Review and Proposed Directions. *Transport Reviews*, Vol. 38, No. 6, 2018, pp. 786–814.
43. Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B (Methodological)*, Vol. 58, No. 1, 1996, pp. 267–288.