

A Visual Analytics System for Exploring, Monitoring, and Forecasting Road Traffic Congestion

Chunggi Lee, Yeonjun Kim, Seungmin Jin, Dongmin Kim, Ross Maciejewski, Senior Member, IEEE, David Ebert, Fellow, IEEE, and Sungahn Ko, Member, IEEE

Abstract—We present an interactive visual analytics system that enables traffic congestion exploration, surveillance, and forecasting based on vehicle detector data. Through domain expert collaboration, we have extracted task requirements, incorporated the Long Short-Term Memory (LSTM) model for congestion forecasting, and designed a weighting method for detecting the causes of congestion and congestion propagation directions. Our visual analytics system is designed to enable users to explore congestion causes, directions, and severity. Congestion conditions of a city are visualized using a Volume-Speed Rivers (VSRivers) visualization that simultaneously presents traffic volumes and speeds. To evaluate our system, we report performance comparison results, wherein our model is more accurate than other forecasting algorithms. We demonstrate the usefulness of our system in the traffic management and congestion broadcasting domains through three case studies and domain expert feedback.

Index Terms—Traffic, Road, Congestion, Visualization, Deep Learning, LSTM, Surveillance, Forecasting, Predictive Analysis

1 INTRODUCTION

Congestion is one of urban transportation's most prevalent challenges, and identifying the true cause of congestion is a major strategic issue in urban planning. As such, many government organizations have invested considerable resources towards the installation of vehicle detectors and Intelligent Transportation Systems (ITS) [1] to analyze and monitor road congestion. However, analyzing and surveilling congestion is difficult due to the high volume and complexity of the data. This issue is further compounded by the lack of effective analytical systems. Analysts in a city traffic management center use vehicle detector data for congestion analysis, but their analysis is mainly performed with table-based statistical tools and simple visualizations (e.g., EXCEL, SPSS). Similarly, reporters in a radio station report on congestion information (e.g., traffic accidents) that has been collected by monitoring hundreds of CCTVs. However, monitoring a vast number of surveillance cameras is difficult, and important information that could affect near-future traffic conditions (e.g., illegal parking, vehicle accidents) is often missed. As such, there is a need for systems that can monitor a variety of streaming data from multiple sources and integrate this information for real-time congestion monitoring, prediction, and retrospective analysis.

Many visualization systems for traffic analysis have been proposed, but they cannot be easily adapted for large cities' real-time traffic surveillance, analysis, and management. Two systems exist for traffic surveillance [2], [3], but they are not scalable for

large city traffic monitoring and analysis tasks, as their focus is on small area surveillance. Other traffic analysis systems [4], [5], [6] can be used for analyzing wider areas, but their traffic data are calculated from GPS trajectories, which is not preferred for real-time surveillance due to computation and time costs. Above all, no existing visual systems provide congestion forecasts in real time, which is important for many tasks such as traffic signal adjustment and radio broadcasting traffic conditions.

In this paper, we present a visual analytics system to support analysts and reporters in the congestion management and traffic broadcast domains. We have conducted interviews with analysts in these domains to derive task requirements and found that the analysts want to see traffic patterns (flow and velocity) and congestion. For this, we provide two views (StreamView and PropagationView) to view the congestion in different perspectives. In the StreamView, we provide Volume-Speed Rivers (VSRivers) that show an entire city's congestion conditions on a map by representing roads' traffic volumes and speeds with shapes and colors. In the PropagationView, we provide Propagation Arrows (PropArrows), a node-link diagram application, to show how congestion propagates on the map with arrow tails and heads representing congestion start and end locations respectively. VSRivers and PropArrows form the basis of our multiple-coordinated views [7] and allow users to 1) explore temporal congestion patterns with estimated congestion propagation, 2) monitor real-time congestion conditions, and 3) understand predicted congestion produced by Long Short-Term Memory models (LSTM) [8]. For evaluation, we present three case studies and provide experts' feedback. To the best of our knowledge, our system is the first system that supports a broad range of traffic analysts (e.g., urban planners, congestion analysts, traffic reporters) by allowing real-time congestion surveillance, forecasting, and analysis over large areas (i.e., city-wide analysis).

Our contributions include: (1) the design of a visual analytics

- Chunggi Lee, Yeonjun Kim, Seungmin Jin, Dongmin Kim, and Sungahn Ko are with UNIST. E-mail:{cglee, yeonjunkim, dryjins, rocky112358, sako}@unist.ac.kr
- Ross Maciejewski is with Arizona State University. E-mail:rmaciej@asu.edu
- David S. Ebert is with Purdue University. E-mail:ebertd@purdue.edu

system with a multi-source traffic data pre-processing pipeline for retrospective analysis, real-time surveillance, and predictive modeling of traffic congestion; (2) the incorporation of the LSTM model for congestion forecasting, and; (3) quantitative and qualitative evaluations of the system with forecasting accuracy comparisons, three case studies, and domain experts' feedback.

2 RELATED WORK

In this section, we describe existing visualization systems for traffic analysis and approaches for short-term traffic forecasting.

2.1 Visual Analysis of Traffic Congestion

Two types of data—location-based and movement-based—have been used for traffic analysis and visualization. Location-based data is collected by vehicle detectors (e.g., laser scanners, video cameras, and inductive loops) [1] that periodically generate average speeds and volume data at installed locations. Two visualization systems exist for laser scanner and video data. Guo et al. [2] visualize traffic patterns of intersections based on laser scanner data, including U-turns and violation patterns. Piringer et al. [3] present AIVIS which combines video data and an incident detection system for road tunnel surveillance. These systems are designed to monitor small areas and are not suitable for large area (i.e., city-wide) surveillance; in comparison, our system is designed for large area surveillance. Inductive loop sensor data, where sensors are installed to cover larger areas, are often utilized for wide area analysis. For visualizing data over large areas, pixel-based visualizations [9] are frequently utilized [5], [10], [11]. Wang et al.s study [5] is similar to our work, as they visualize both traffic speed and volume data on a map. To visualize the data, they encode flow directions with angular brackets and speed data with colors within a symbol. However, this encoding may cause confusion due to unclear flow directions and occlusions among the symbols. In contrast, we dynamically adjust our system's visualization components according to roads volume data and zoom level to prevent occlusion. Furthermore, previous visual analytics do not provide real-time congestion forecasts, whereas our system is specifically designed for forecasting.

Movement-based data (i.e., object movement data collected via GPS) has also been utilized in traffic analysis and visualization for monitoring [6], [12], [13]. Route zooming [13] non-linearly adjusts road sizes to directly embed temporal data on broadened roads. However, such zoomable interfaces may not be ideal when users need to explore and compare a large number of roads (431 roads across the city are monitored in our case study examples). Our system reduces the need for zoom operations by visualizing temporal data in the linked views and only presents roads volume and speed data with VSRivers, allowing for quick comparisons and overviews of roads.

Many visual analytics systems have focused on traffic routing due to the ubiquity of GPS data, and GPS movement data are often combined with other data such as transpiration or urban data [4]. For example, VAUD [14] uses movements from GPS to enable the visual exploration of cross-domain data (e.g., social networks, microblogs, trajectories, real state data). Other studies have focused on transportation visualization [15], [16], [17], [18], [19], [20], [21], and we refer the readers to recent surveys on the topic [22], [23], [24]. Many of these traffic visualization techniques rely on GPS data. However, GPS data for traffic analysis requires calculating vehicle speed and volume [4], and

these calculations are often unable to support real-time congestion analysis due to computational costs.

2.2 Short-term Traffic Forecasting

While much of the previous work in the visual analytics community has focused on retrospective traffic analysis, a variety of models have been developed for traffic forecasting and simulation. For short-term traffic forecasting [25], most methods can be categorized as either parametric (model-based) or non-parametric [26]. In the parametric approach, parameters need to be computed with empirical data for pre-defined analytical models. Examples in this category include time-series analysis techniques (e.g., ARIMA [27]). The non-parametric approach does not assume any models or parameters, as it assumes that models should be derived from data. Thus, more data is required in the non-parametric approach than in the parametric approach (i.e., data-driven approach). Most machine learning techniques are included in this category, such as support vector machines (SVM) (e.g., [28]) and artificial neural networks [29]. Some studies have reported that the non-parametric approach outperforms the parametric approach in many cases, particularly with neural network models [29]. One cause of this difference is that the parametric approach assumes stationary and linear data, which is not guaranteed (e.g., rush hour or accidents). The non-parametric approach does not have such assumption and can learn non-linearity in data [29].

The recent success of deep learning in many domains [30] has resulted in a variety of research utilizing deep learning models for traffic forecasting. Examples include the experiments with Deep Belief Networks (DBN) [31] and the stacked autoencoder (SAE) for learning road network features from volume and speed data [32], [33]. Although more accurate than conventional models, these models share a limitation in that spatiotemporal relationships (e.g., spatial dependency in road networks, non-linear temporal dynamics with changing road conditions) are not considered.

Other studies have explored methods for capturing spatiotemporal relationships in traffic data [32], [34], [35], [36], [37], [38]. Although a number of architectures have been proposed, the Recurrent Neural Networks (RNNs) and Long-Short Term Memory (LSTM), a variant of RNNs, are the most popular due to their ability to model temporal dynamics [36], [37], [38], [39]. Convolutional neural networks (CNNs) are another method used for traffic forecasting, but when they are used, another process of extracting road networks and generating graphs from images is required (e.g., GCNN in [40]). Other work has focused on a mixture of different models. Li et al. [38] demonstrate that combining diffusion convolutional recurrent neural networks (DCRNN) with graph convolution [41] can generate accurate short-term traffic forecasting. A hybrid of LSTM combined with a CNN is also proposed [35], where the CNN's main role is to learn the spatial dependency of traffic from images. We adapt the base LSTM model to capture temporal dynamics in the speed data sequences and embed a road network structure produced by DeepWalk [42]. Compared to prior work, our model is the first one based on LSTM that learns not only spatiotemporal road speed and network features, but also rush hour information to improve accuracy.

3 TASK DESCRIPTION AND REQUIREMENTS

Our system has been designed based on input from an expert from a congestion management center and three experts from a congestion broadcasting station in the city of Ulsan, South Korea,

where more than 1.1 million people live as of 2017. We have met regularly with the experts for 18 months to extract task requirements, discuss design considerations, and demonstrate prototypes in progress to collect feedback. We started our collaboration with a semi-structured focus group interview. The first group of experts consisted of an assistant director (PhD in transportation, 20+ years experience), a manager (task: congestion analysis, 18+ years experience), and one radio reporter (task: congestion reporting, 3+ years experience) from a regional branch of a nationwide radio broadcasting center.

According to our experts, the center's main mission is to provide timely and accurate information on congestion for drivers. To fulfill their mission, traffic information (e.g., traffic congestion, vehicle accidents, etc.) is collected 24 hours a day by receiving phone calls from non-paid volunteers (e.g., taxi drivers) and monitoring 128 CCTVs. Congestion is reported 6–8 times during rush hours, during emergency situations (e.g., vehicle accidents), and 4 times per hour during off-peak times.

One of the center's issues stems from a recent decrease in the number of unpaid volunteers and human resource personnel assigned to monitor CCTV. To overcome the issue, our partners have started using a system that provides real-time speeds with colored lines on a map [4]. However, the visualization from the website is not useful, as it lacks data on vehicle volumes on roads. According to the experts, volume information is critical, as a large volume means more vehicles on the road, and sudden changes in volume often represent important events such as accidents. Since their top priority is to broadcast urgent traffic situations, they need a means for identifying urgent roads at each broadcast. The next issue is that although they can initially detect congested roads by watching CCTV, they do not have any method for estimating the propagation of the congestion. Thus, they do not broadcast information on congestion propagation, which could help drivers choose a detour. Finally, there have been many requests for forecasting short-term traffic conditions, but our partners do not have any solution for this, as conventional traffic forecasting methods have not produced reliable results in their tests.

The group leader of the city traffic management center (PhD in traffic management, 10+ years experience) also reported the necessity of an effective means for analyzing and monitoring congestion. According to him, the center mainly focuses on two tasks: 1) monitoring and analyzing congestion patterns and changes, and 2) maintaining the intelligent transportation system (ITS) infrastructures. He also mentioned that the first task involves a city congestion improvement project, in which a congestion optimization effort (e.g., traffic signal optimization) is devoted to severely congested roads. The second task is related to fixing vehicle detectors so that not only his group, but also police officers in the same building, can better monitor the impacts of their traffic signal control. In order to decide when detectors need replacing, the analysts must analyze the detectors operation time and accuracy data, as replacing installed detectors requires expensive road construction. Currently, such investigations are hard to perform frequently due to the number of detectors across the city, as shown in Fig. 1. Currently, the center lacks an efficient way to complete its duties—it uses a tool that computes volume and speed statistics, but all the computed results, as well as the raw data, are presented in Excel with simple charts, which is insufficient for performing its tasks.

From our task analysis, we have derived the following requirements for a system for domain experts:

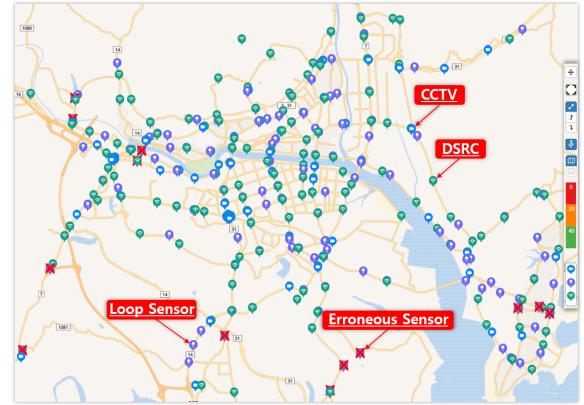


Fig. 1. An overview of installed vehicle detectors and CCTVs across the city. 'X' marks are placed on the sensors when more than 50% of produced data over 6-month are erroneous (e.g., negative values).

- (R1) Analysis of congestion patterns, changes, and trends with historical data;
- (R2) Real-time congestion surveillance across the city;
- (R3) Real-time congestion propagation estimation;
- (R4) Real-time predictive analysis of near-future congestion conditions, and;
- (R5) Real-time maintenance of malfunctioning vehicle detectors.

We describe research problems related to the requirements and a summary of our methods and interfaces as follows.

(Prob1): How can we visualize congestion conditions across a city for effective congestion analysis and efficient surveillance (R1, R2)? For this problem, two design decisions are available: 1) embedding all the information into a map [13], or 2) presenting the minimum amount of information on the map for congestion analysis and loading auxiliary data in multiple coordinated views (MCVs). In Sec. 5.2, we discuss the pros and cons of each choice and present VSRivers, a visualization for congestion analysis.

(Prob2): How can we forecast road speeds and congestion conditions (R4)? By having congestion forecasts, analysts can promptly start visual congestion analysis with our system and take early actions to reduce the impact of the forecasted congestion. Based on our literature survey, two main options are available between the LSTM and CNN models. In Sec. 4, we discuss the pros and cons of the models and how we have trained the LSTM with spatiotemporal and rush hour features for improved accuracy.

(Prob3): We need a definition and method for detecting congestion and its propagation for visualization (R2, R3). We describe our approach to this problem in Sec. 5.1 and Sec. 5.2.2.

(Prob4): How do we facilitate real-time congestion analysis and surveillance tasks (R2-R5)? Prior work calculates congestion data from trajectories [4], but this approach is not ideal due to its computational costs. To address this problem, we collect vehicle volume and speed data in real time from ITS sites. We present our processing pipeline and model training process for forecasting short-term congestion (R4) in Sec. 4.

4 DATA PRE-PROCESSING AND MODEL TRAINING

We describe our pre-processing pipeline (Prob4) and model training process for forecasting short-term congestion (R4). We have used dedicated short range communications (DSRC) data [43] (range: 9/1/2017-2/28/2018) and loop sensor data [44] (range:

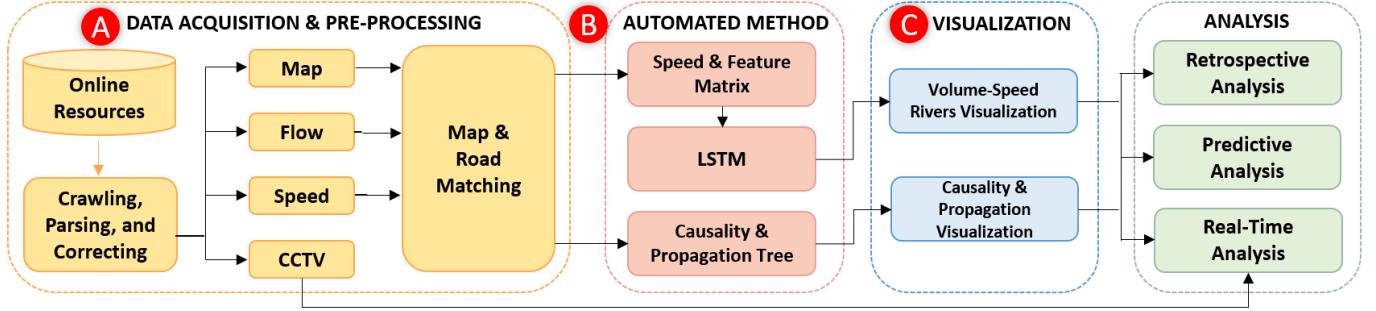


Fig. 2. An overview of the system workflow with functional modules. (A) Data acquisition & Pre-processing, (B) Automated methods for computing congestion points and forecasts, and (C) Visualization module for VSRivers and congestion propagation.

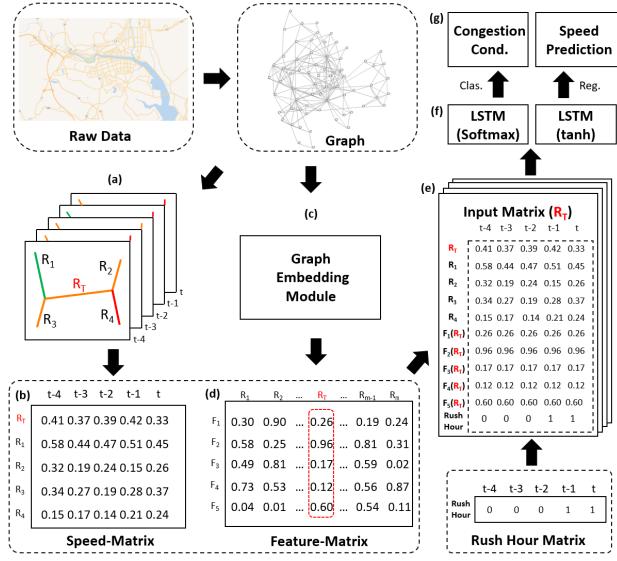


Fig. 3. Illustration of our LSTM training process. From raw data (a), spatiotemporal road dependency is extracted in (a) and (b), while latent features among roads are computed in (c) and (d). The final input matrix contains speed and network features from both matrices.

1/1/2016-2/28/2018) from Ulsan, South Korea, where 1,164,489 citizens live, and 549,489 vehicles are registered as of January, 2018. There are 189 DSRC vehicle detectors installed across the city approximately 5.7km apart, covering 68 main roads. The city maintains a public ITS server that updates information every minute (data format: road name, speed [km/h], road start and end locations). There are 201 inductive loop sensors installed in the city approximately 1.5 km apart, covering 431 roads (i.e., denser than DSRC data). Data from the loop sensors are aggregated at 15 minute intervals according to an official guideline [45] and posted daily on the public ITS web site (data format: road name, start and end locations, direction, speed [km/h], volume). We use the DSRC's speed data to enable real-time congestion analysis and prediction and loop sensors' volume to visualize roads' average volume in our VSRivers (Sec. 5.2.1). Detector and CCTV locations are presented in Fig. 1. After several discussions with the domain experts, we have decided to use both DSRC and loop sensor data to support real-time surveillance and forecasting. Fig. 2 (A) describes our pre-processing steps.

Two main modeling options are available to predict road congestion: CNN and LSTM. After reviewing the two models, we chose to adapt the LSTM model because it can forecast road

speeds and congestion conditions that experts can use. LSTM has also demonstrated the ability to learn non-linearity from sequences (e.g., consecutive speed data on roads) in many studies [36], [37], [38], [39]. The CNN model can also be used (e.g., [40]), but it does not predict the speeds of roads, which the reporters need.

We present an overview of our model training process in Fig. 3. To train the model, one must provide appropriate features to the model in the form of a matrix. For this, we extract road segments from intersections and DSRC vehicle detector locations and build spatiotemporal speed matrices for target roads as follows:

$$x_{t+1}^{R_T} \Leftarrow \begin{bmatrix} x_{t-n}^{R_T} & x_{-(n-1)}^{R_T} & \dots & x_{t-1}^{R_T} & x_t^{R_T} \\ x_{t-n}^{R_1} & x_{-(n-1)}^{R_1} & \dots & x_{t-1}^{R_1} & x_t^{R_1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{t-n}^{R_{\delta-1}} & x_{-(n-1)}^{R_{\delta-1}} & \dots & x_{t-1}^{R_{\delta-1}} & x_t^{R_{\delta-1}} \\ x_{t-n}^{R_{\delta}} & x_{-(n-1)}^{R_{\delta}} & \dots & x_{t-1}^{R_{\delta}} & x_t^{R_{\delta}} \end{bmatrix} \quad (1)$$

Here, t is the current time step, R_δ represents the neighboring roads of the target road (R_T), and $x_{t+1}^{R_T}$ is the forecasted speed of the target road R_T at $t + 1$. Each element in the matrices is the normalized speed of each neighbor road R_δ at each time step to speed up the training time [46]. The rows in the matrices represent different neighbor roads, while the columns represent different time steps. In order to represent a target road in training, we consistently put the target road data into the first row of the matrices. In our experiment, a random distribution of target roads in the matrix resulted in a less than optimal performance. Below the target road data row, we locate the four shortest roads (by length) among all the linked roads on the target road. When there are fewer than four linked roads, we place the target road data to fill in empty rows. In our experiments, using more than four roads did not improve the forecasting performance. We use five time steps with 5-minute intervals in the columns. Note that we set the time interval for output at 15 minutes based on suggestions from the domain experts. Fig. 3 (a) shows an example road network where its target road R_T is linked with four other road segments (R_1-R_4 , $\delta=4$) and is placed in the first row of the input matrix in Fig. 3 (b). We call this model LSTM (T) to represent that this model uses temporal speed data only.

Next, we extract latent similarity features (e.g., correlations regarding congestion occurrence times and locations) among roads for training by embedding a road network into low-dimensional vector space (Fig. 3 (c, d)). To do so, we employ DeepWalk [42]

with road nodes and links as inputs. DeepWalk is a graph representation learning method that overcomes the computational cost of graph embedding. We refer the reader to Goyal and Ferrara's [47] and Cai et al.'s [48] surveys for detailed explanations and performance comparisons of graph embedding methods. We extract five network features for each road from DeepWalk and create a feature matrix as shown in Fig. 3 (d) with the columns representing different roads (R_1-R_m) and the rows representing extracted features (F_1-F_5). In our experiments, using more than five features did not improve the forecasting performance. Next, inspired by Li et al.'s work [38], we merge the speed and road similarity feature matrices into the final 10×5 spatiotemporal matrix (Fig. 3 e) with the five columns being different time steps and the ten rows being speeds and similarity features. Note that the same feature values of a target road are repeatedly used for different time steps because road spatial features (e.g., road connectivity) are invariant over time. We refer to this model as LSTM (S,T) to represent that this model uses both spatial road similarity and temporal speed data. Finally, we provide the model with information about rush hour (6-9am, 4-8pm) as shown in Fig. 3 (h), where 1 means rush hour time. We refer to this model as LSTM (S,T,R) to distinguish the final model from the aforementioned models.

The LSTM model can be used for predicting both road speeds and congestion conditions (Fig. 3 g) by using different activation functions in the last layer (Fig. 3 f). The model using tanh as its activation function in the last layer produces predicted speeds for each road. The model using softmax as its activation function in the last layer generates the probability of each condition for each road and then selects the condition with the highest probability (i.e., argmax). Because our model has different activation functions for the tasks, we use two optimizers—we use the mean square error (MSE) for the regression task and categorical cross entropy [49] for the congestion condition classification task. Table 1 and Table 2 in Sec. 6 present the speed and congestion condition prediction performance, respectively.

In our experiment, we use 80% of the data for training, and the rest of the data is used for testing. The validation is performed with 20% of the training data. We use Adam [50] for learning rate optimization and perform a grid search for hyperparameter optimization—4 stacked LSTM layers with 100 units, a dropout of 0.1, a batch size of 128, an L1 regularization of 0, an L2 regularization of 0, and a learning rate of 0.001. We use dropout [51] and early stopping [52] with validation accuracy. The DeepWalk parameters are a walk number of 80, a representation size of 5, and a walk length of 40. Note that all the parameters are produced with our data and may be different for other congestion data.

We run 50 epochs with the hyperbolic tangent function as an activation function in training and a softmax function as an activation function in the last LSTM layer. Each epoch takes 160 seconds. The size of the final merged matrices used in training is 770MB for 107 days. We use a server for training (CPU: Intel Xeon E5-2640, 2.6GHz, 64GB RAM, NVIDIA GeForce GTX1080 Ti 12GB RAM). In terms of scalability for forecasts with streaming data, our system retrieves 24KB of text data of road names, stream directions, and speeds from a public ITS database every minute. Then, the data is parsed in 0.16 seconds and forecasts are produced in 0.6 seconds in our server (Intel Xeon CPU E5-2630, 2.40GHz, 128GB RAM), which is enough for real-time forecasting. The crawling, visualizations, and forecast with streaming data are performed and provided by our server.

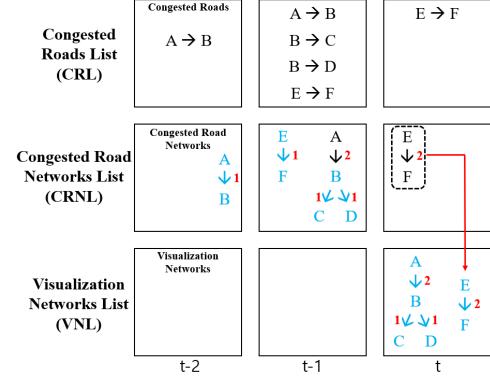


Fig. 4. An example scenario for detecting congestion causes and propagation directions. At ($t - 2$): Initially, congested roads are in the CRL and CRNL. At ($t - 1$): Whenever new congested roads are detected, they are added to CRNL after the road network among the congested roads is built with increased severity weights. At (t , present): CRL and CRNL are updated again. Then, the networks in VNL are visualized.

5 VISUAL ANALYTICS ENVIRONMENT

In this section, we describe our visual analytics environment. We first present our automated method for detecting congestion events (R3). Then we describe our visual interfaces, as shown in Fig. 2 (C) that includes Volume-Speed Rivers (VSRivers), Propagation Arrows (PropArrows), and visualizations in multiple coordinated views for spatiotemporal analysis.

5.1 Congestion Propagation Estimation

In order to help experts (e.g., congestion reporters in a radio broadcasting center), whose tasks involve estimating congestion propagation (R3), we use a method for detecting congestion propagation events. First, we define a congested road R at time interval T as $[R, T]$. Then, we represent congestion propagation from R_n to R_{n+1} as $[R_n, (T_x, \dots, T_k)] \rightarrow [R_{n+1}, T_s]$. Here, we assume that R_n and R_{n+1} are physically linked and $x < s \leq k$ to imply that R_{n+1} 's congestion starts only when R_n is still congested. The assumptions also mean that we only detect backward propagation according to our domain experts experience, observations, and research experiments [53], [54].

In order to compute the severity of congestion on road (s)—an estimated waiting time on the road due to congestion, we use a weighting method where a road with a longer congestion time has a higher weight. An example scenario of our congestion detection, propagation, and weighting method is presented in Fig. 4. Here, we observe at $t - 2$ that road AB is initially congested, where A and B are the start and end locations of AB respectively. When AB is initially detected, our system registers it on the congested roads list (CRL) and searches linked roads of AB . As there is no linked road to AB , the congested road networks list (CRNL) is the same as CRL. Note that newly added networks are presented in blue in Fig. 4. Next, we observe at $t - 1$ that other roads (BC , BD , and EF) start to become congested, while AB is still congested in the CRL. Here, our system investigates the city's road network again and builds congested road networks in the CRNL. Arrows in the CRNL are used to represent congestion directions, while the numbers in red assigned to the arrows represent the congestion weight of the congested roads. Also, the congestion weight of AB becomes 2 at $t - 1$, because AB has existed in the CRNL list for two time steps (i.e., $t - 2$ and $t - 1$). At time step t (i.e., present), previous

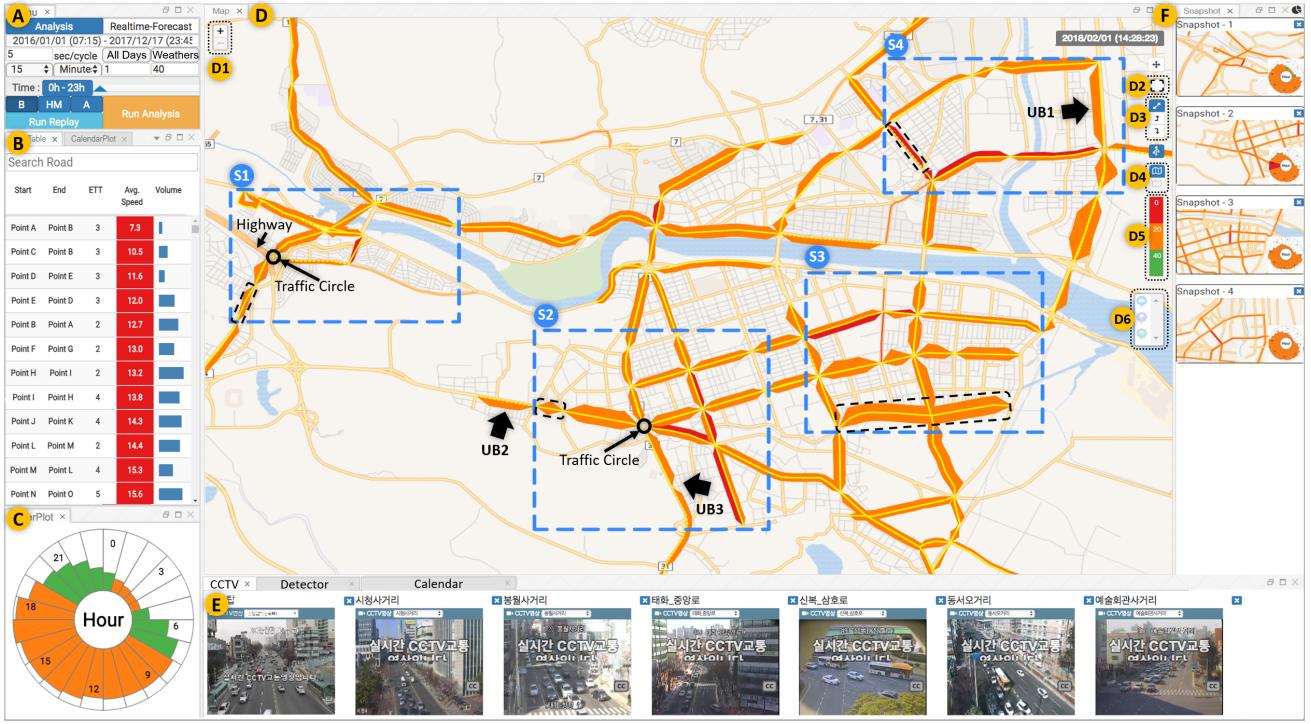


Fig. 5. An overview of our system. (A) A view for filtering by date ranges, day of week, traffic volume, traffic speed, and hour. (B) A sortable table view for displaying estimated travel time, average speed, and volume for each road, (C) A clock view for seasonality analysis. (D) A map view with Volume-Speed Rivers Visualization. (E) A view for real-time road monitoring with CCTV streams (CCTV tab), pixel-based detector visualization (Detector tab), and calendar visualization (Calendar tab). (F) A snapshot view for monitoring important roads in real-time.

roads are no longer congested, while EF is still congested. Then, the previously congested roads, as well as the currently congested road EF in the CRNL, are moved to the visualization networks list (VNL) for visualization. With the weight sets of roads (A-B-C, A-B-D, E-F) at time step t in the VNL, we define the congestion severity, S , as follows:

$$S = \frac{\sum_{i=0}^n (\text{CongestionWeight}_i * \text{interval})}{n} \quad (2)$$

Here, n is the number of roads, and we set the *interval* between $t-1$ and t as 15 minutes in Fig. 4.

5.2 Visual Interfaces

Two design approaches exist for visualizing traffic data for congestion analysis. The first approach is to present as much data as possible on a map while a user explores the information using zoom operations. Route-zoom [13] is a technique within this approach that non-linearly adjusts road sizes to embed sufficient data into a map. Since this approach assumes “zoomed-ins” before adjusting a road’s size, only a small number of roads and their congestion conditions are simultaneously visualized and compared. Thus, many zoom and pan operations are required when exploring a large number of roads (431 roads in this work). Another design approach is to utilize MCVs with a map presenting the most important data for congestion exploration and multiple views representing auxiliary data for temporal analysis, allowing for quick congestion comparisons and overviews. A possible disadvantage of this approach is a longer analysis time compared to the first approach [13]. Reviewing the two approaches, we utilize MCVs because they have proven effective in prior work [4], [6] in reducing the amount of interaction to get to an analysis result

and possible analysis errors during exploration [55] and providing overviews of congestion conditions for simultaneous comparisons among congested roads.

Our system utilizes multiple-coordinated views [7] to represent the spatiotemporal aspects of the congestion data. There are ‘analysis’ and ‘realtime-forecast’ modes in our system. In the analysis mode, users can load congestion data for retrospective analysis (R1) and filter by date, time, day of week, weather (e.g., sunny, rain, snow, cloudy), speed, and volume of roads, as shown in Fig. 5 (A). In the realtime-forecast mode, users can select either the real-time or forecast mode. When the forecast mode is chosen, the predicted road speed and congestion condition data provided by the LSTM are used in the system. When the system is not in the forecast mode, congestion conditions are derived by identifying speed ranges of the raw data. Animating (or replaying) past congestion conditions with a specific time range and interval is also supported. The table view in (B) presents a road list, where roads can be sorted by speed, volume (i.e., bar chart), and estimated travel time (ETT, minutes) per road. The clock view in Fig. 5 (C) shows hourly-averaged speeds with different colors (green: unimpeded, orange: slow, red: congested) and volumes with radii. The calendar view, Fig 8, is shown when the ‘Calendar’ tab is clicked. The calendar view shows both upstream (left) and downstream (right) daily-aggregated speed and volume data. Note that the bars in the calendar view are presented at the end of each row and column for comparing congestion aggregated by week or day of week.

Our system provides two interaction methods for real-time congestion surveillance. The first one is that a user can click a CCTV icon on a map and watch individual CCTV streams as seen in Fig. 5 (E). Additionally, a user can select an area, and

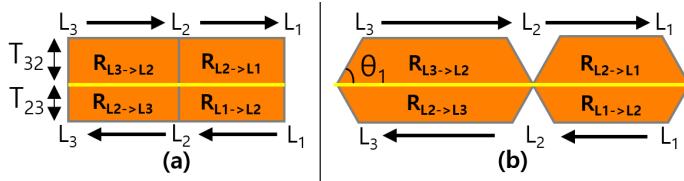


Fig. 6. Illustration of Volume-Speed Rivers (VSRivers) visualization that spans three vehicle detectors (L₁–L₃). Arrows represent the direction of vehicle flow (i.e., upstream and downstream), and a yellow line means the double yellow line on roads. Volume and speed data of a road segment, R_{L3-L2} are mapped to the thickness (e.g., T_{32}) and color (orange), respectively.

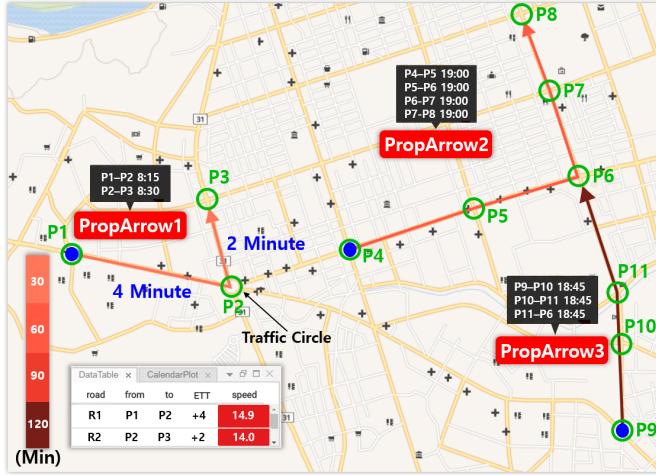


Fig. 7. An example PropArrows visualization that presents blue circles as root causes, propagation directions (i.e., arrow heads), and severity (i.e., colors).

its roads are kept in the snapshot view for monitoring groups of roads in real-time, Fig. 5 (F). Currently, 4 selected areas are shown. Six filtering options are available in the map view in Fig. 5 (D), including zooming (D1), enabling area selection (D2), filtering by stream directions (D3) and congestion conditions (unimpeded (green): faster than 40-km/h, slow (orange): 20-40km/h, impeded (red): 0–20km/h (D5), adjusting background opacity (D4), and presenting CCTV cameras, loop, and DSRC detector icons on the map (D6). In the realtime-forecast mode, streaming data is fed into the table view (B) and map view (D) for real-time monitoring (R2). For R5, we incorporated a pixel visualization and provide DetectorVis, Fig. 9, to visualize speed data from sensors with the x-axis representing hours (0–23) and the y-axis representing different dates.

Our system is designed so that a user interaction in one view changes other views' visual states. For example, when a congested road in (B) is clicked, the information of the road is presented in the clock and calendar views, and the road is highlighted in the map view. When a time in (C) or a date in (E) is selected, other views visualize the congestion conditions of the selected time or date. Next, we introduce VSRivers and PropArrows visualizations to view the data in different perspectives. Note that these two visualizations are not used in combination in one visualization.

5.2.1 StreamView: Volume-Speed Rivers Visualization

After a series of discussions with our analysts, we identified two important requirements of a visualization for congestion analysis.

First, the visualization should provide not only overviews of congestion conditions across the city, but also individual road congestion conditions without requiring frequent user interactions (e.g., zoom operations). Second, the visualization should help analysts quickly identify the condition of important roads. Here, important roads are the roads with larger volumes and slower speeds. Given these requirements, we have designed Volume-Speed Rivers (VSRivers) visualization that embeds traffic volume and speed attributes into maps. Our VSRivers are shown in Fig. 5 (D). Our initial VSRiver is as shown in Fig. 6 (a). Here, L_x presents locations of vehicle detectors. Volume and speed data of a road segment, R_{Lx-Ly} are mapped to the thickness (e.g., T_{32}) and color (orange), respectively. In selecting colors for representing congestion conditions, we use conventions in the domain, that is, green for a speed faster than 40km/h (i.e., unimpeded), orange for a speed slower than 40km/h (i.e., slow), and red for a speed slower than 20km/h (i.e., impeded).

Using the initial version in Fig. 6 (a), we soon found that when a road strip (i.e. a long road) with multiple roads have the same congestion condition, we cannot detect the number of roads in the strip. In order to resolve this problem, we revised the initial version such that a boundary between two roads is distinguished by dropping the thickness at each end. This revised version is shown in Figure 6 (b), where θ is the angle that decides a dropping point. A default value of θ is 30 degrees, but it changes up to 22 degrees, according to the road type and optimization in the map-matching process. For example, for 'V'-type roads, 22 degrees of θ is used to avoid occlusion with an early drop. For an analysis of wide road networks with VSRivers, we define VSRivers as a projection of all individual VSRiver. Note that when a road segment length is too short for visualization, the VSRiver on the road becomes a triangle. In this case, the thickness can be remained to present the accurate volume information with adjusted θ values. When the triangle encoding is used, details are small and clutter is inevitable, but zooming-in fixes this. When sufficient space for the road is allowed (e.g., zoom-ins), the triangle becomes a VSRiver on the road (i.e., the level of detail design approach). We normalize the thickness of VSRivers among parallel roads to prevent occlusion. Note also that streaming data are fed into VSRivers in the realtime mode (R2), while forecast data are fed into VSRivers in the forecast mode (R4).

5.2.2 PropagationView: Propagation Arrows Visualization

In order to support the identification of real-time congestion causes and propagation directions (R3), we have proposed a heuristic method in Section 5.1. The method produces three types of information on congestion: start and end locations of congested roads; propagation directions (i.e., arrows in Fig. 4), and; duration (i.e., weights around the arrows in the road networks in Fig. 4). To visualize the information, we utilize color-coded propagation arrows (PropArrows) with arrow heads pointing in the direction of congestion propagation and the color of the arrow represents the duration of the congestion. An arrow is visualized along congested roads by using geographical coordinates of the roads [56]. When multiple propagation directions are detected, multiple arrows are presented to describe the event. An example visualization is presented in Figure 7, where many consecutive roads are congested in a day (i.e. retrospective mode), and PropArrow1, PropArrow2, and PropArrow3 represent information on the congestions. Note that this visualization also works in the real-time mode. In the retrospective mode, users can specify a time range, while the previous hour is considered in the real-time

mode for the computation. Start and end locations of congested roads (i.e., P1, P2, ... P7) and congestion start and end times are also shown in tooltips. Note that locations with initial congestion are shown with blue circles (P1, P4, and P9). When a series of linked roads are simultaneously congested (e.g., PropArrow2 and PropArrow3 in Fig. 7), we consider the vehicle flow direction to identify the initial congested road. A legend with four intervals is used for presenting congestion duration times. We merge multiple separated PropArrows into one PropArrow, according to a zoom level, when more than two arrows are shown on a road. To represent that the merged PropArrow is an aggregation result, we color the start location of the PropArrow in green.

From the example visualization in Fig. 7, we see in *PropArrow1* that congestion first started in the road between P1 and P2 at 8:15am and then the road between P2 and P3 was congested after 15 minutes. If the congestion in *PropArrow1* finished at 8:45am, the congestion between P1 and P2 lasted for 30 minutes (i.e., `congestionWeight=2`), and the one between P2 and P3 lasted for 15 minutes (i.e., `congestionWeight=1`). Therefore, the severity, S of *PropArrow1* is $22.5 (2*15+1*15)/2$ minutes according to Eq. 2.

5.3 Implementation Notes

Our system and modules are implemented using several libraries and tools (Fig. 2). For the visualization module, “D3.js,” “Django,” and JSON are utilized. In order to allow a flexible view arrangement and an interactive map, OpenRouteService, “Golden-layout” and “Leaflet” are used. All the computations, including congestion forecasting in the automated method module have been performed with Python and “Keras.” Intersections and roads are extracted with “Shapely” in the pre-processing module.

6 FORECASTING PERFORMANCE EVALUATION

In order to quantitatively evaluate the model performance, we have performed two experiments with 3-months of DSRC speed data (2.7M records, 5 minute interval). The first experiment is conducted using a conventional short-term traffic forecasting algorithms [25], including autoregressive integrated moving average (ARIMA) [27], [57] with the Kalman filter, Support Vector Regression (SVR) [58] and the LSTM (T) [36] and LSTM (S,T). Among the methods, LSTM (S,T,R) demonstrates the best performance in road speed forecasting, as shown in Table 1. Specifically, the LSTM (S,T,R) shows a 31.5%, 19.4%, and 34.0% improved MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and MAPE (Mean Absolute Percentage Error) over SVR. LSTM (S,T,R) produces better results than previous LSTM versions (e.g., a 4.0%, 3.7% and 5.9% improved MAE, RMSE, and MAPE over the LSTM (T), a model in used in [36]).

The next experiment determines whether LSTM (S,T,R) can be used for forecasting congestion conditions (i.e., slow, impeded, and unimpeded). In this experiment, we employ conventional classification methods, multi-layer perceptron (MLP), Support Vector Machine (SVM), and decision trees (DTs) with the flattened feature vectors in Figure 3 (e) for comparison. We use precision, recall, and F_1 measures to quantify classification performance. Our experimental results (Table 2) indicate that the LSTM (S,T,R) shows a 7.5% improvement in F_1 over DT, but demonstrates a similar performance when compared to MLP and LSTM (T) in slow conditions. However, LSTM (S,T,R) demonstrates the best result in impeded conditions, which are a more important type of

Task	MAE	RMSE	MAPE
ARIMA _{kal}	0.0644	0.0913	16.4079
SVR	0.0549	0.0746	13.9120
LSTM (T)	0.0392	0.0624	9.7548
LSTM (S,T)	0.0383	0.0611	9.3670
LSTM (S,T,R)	0.0376	0.0601	9.1779

TABLE 1
Our LSTM (S,T,R) demonstrates the best performance in forecasting the road speed compared to conventional forecasting algorithms.

Task	Slow Condition			Impeded Condition		
	Prec.	Recall	F_1	Prec.	Recall	F_1
SVM	0.695	0.662	0.678	0.361	0.460	0.404
MLP	0.862	0.885	0.873	0.807	0.670	0.732
DT	0.825	0.826	0.825	0.661	0.690	0.675
LSTM (T)	0.888	0.864	0.876	0.741	0.721	0.731
LSTM (S,T)	0.888	0.873	0.880	0.776	0.768	0.772
LSTM (S,T,R) _R	0.867	0.901	0.884	0.813	0.739	0.774
LSTM (S,T,R)	0.895	0.880	0.887	0.805	0.772	0.788

TABLE 2
LSTM (S,T,R) shows the best improvements in forecasting impeded conditions, compared to the conventional classification methods. LSTM (S,T,R)_R derives congestion conditions from the predicted speeds.

forecasting than slow conditions. Specifically, the LSTM (S,T,R) shows a 7.6%, 16.7%, and 7.7% improvement in F_1 values over MLP, DT, and LSTM (T), respectively. We see that the LSTM (S,T,R) demonstrates a higher F_1 score than LSTM(S,T,R)_R that derives congestion conditions by identifying the ranges of the regression results. Using R improves the performance, but does not have a large impact over the LSTM (S,T).

7 CASE STUDIES

In order to evaluate our system, we describe three case studies. In the first study, we illustrate how an analyst used our system to analyze congestion characteristics of the city. Next, we demonstrate the usefulness of our system for the city’s congestion improvement projects. The final study illustrates how congestion reporters in a radio station utilize our system during their broadcasts.

7.1 Understanding City Traffic Congestion Patterns

An analyst in a congestion management center was asked to report characteristics of the city’s road network and congestion conditions in 2017. To perform the task, the analyst uses the retrospective mode in the system and selects the full range of 2017. Once data is loaded, the analyst first looks at the calendar view that shows the daily-aggregated congestion conditions across the city with different colors assigned to each day, Fig. 8 (a) upstream and (b) downstream. Note that the x-axis and y-axis mean days of week (i.e., Monday–Sunday) and different weeks, respectively. The number at the top represents months. Holidays are shown with black outline boxes and average volumes are shown at the end of each row (i.e., average volumes by day of week) and column (i.e., average volumes by week)

From the calendar view, the analyst first notices that the downstream roads show more orange pixels than upstream roads. This difference is of interest because traffic tends to be balanced (e.g., commuting traffic in the morning and evening). However, as she hovers over pixels, the tooltip reveals that the congestion is not very skewed as speed differences are within 3-4km/h of the average speed of 40km/h (e.g., 40.58km/h and 38.19km/h, as

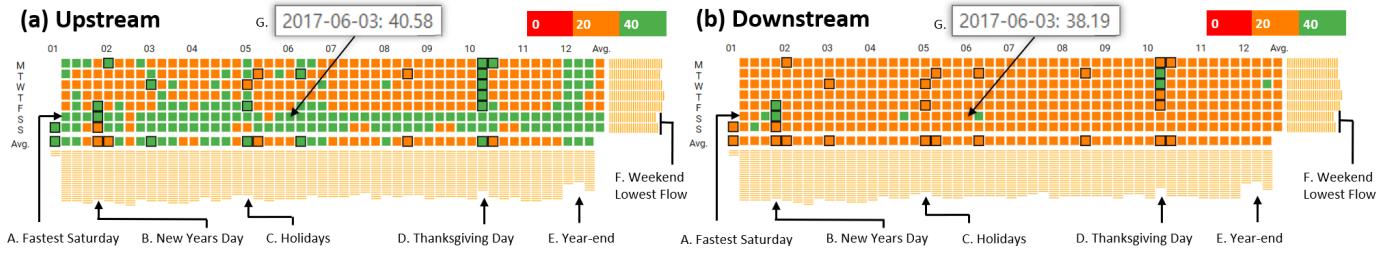


Fig. 8. The calendar view shows an overview of daily congestion conditions in 2017 (x-axis: month, y-axis: day of week). Congestion happens periodically, and there is less congestion during holidays and weekends.

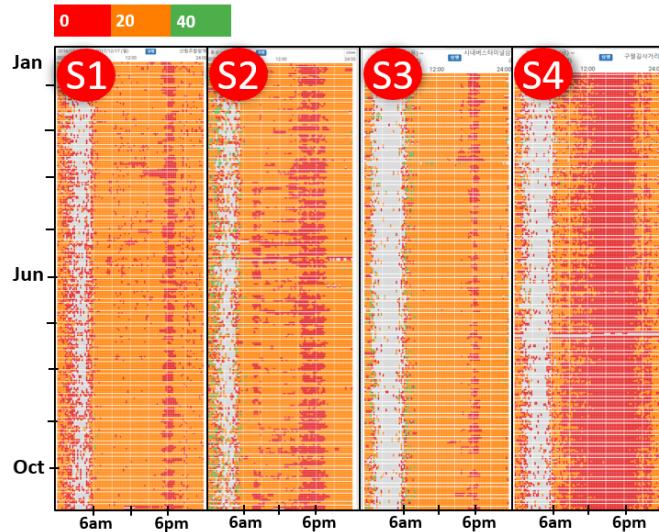


Fig. 9. Roads in the black dashed boxes in Fig. 5 (D) S1–S4 are selected for detailed congestion analysis with DetectorVis.

shown in Fig. 8 G). Note that 40km is the threshold between slow (orange) and normal (green) conditions. Next the analyst looks at the volume bars at the right and bottom parts of the view and notices that less congestion exists during holidays and weekends in both streams (Fig. 8 A–F). To find hourly congestion patterns, she looks at the clock view, Fig. 5 (C) and checks congestion conditions of each stream by using the stream filtering option in Fig. 5 (D3) and notices that most roads in both streams in the city were slow from 8am to 7pm. Understanding an overview of congestion patterns, she decides to further use our system to find detailed geographical congestion patterns.

The next investigation point is the map view, where VSRivers in the StreamView present congestion conditions across the city on the map, Fig. 5 (D). At a glance, she observes 4 areas (S1–S4 in Fig. 5 D) with impeded conditions (S1, S2, S4) or large volumes in the slow condition (S3, black dashed line). To perform an in-depth analysis, she selects a number of roads in the areas (i.e., Fig. 5. S1-S4) and explores the congestion conditions from DetectorVis. Fig. 9 shows the DetectorVis with the y-axis representing dates and the x-axis representing hours (i.e. 0-23). In the visualization, many red lines are observed, meaning many roads tend to be impeded at specific time ranges. S4 was impeded most of the time.

During an exploration of the map view, she notices from VSRivers that several roads have unbalanced vehicle volumes and speeds, such as UB1–UB3 in Fig. 5. For example, she sees on

UB1 in S4 that the downstream road has a higher volume and is 17km/h slower compared to the upstream road's volume and speed. This is interesting to her because traffic on roads is assumed to be balanced if there is no problem, and such unbalanced traffic has not yet been reported. After exploring the map view with extensive zooming and panning around the road, she formulates a hypothesis that the lack of balance in UB1 might have been caused by an overpass. From her point of view, while the overpass provides benefits to upstream traffic (i.e., reducing congestion and traffic volume), it directs downstream traffic to an intersection, contributing to congestion. She notes this and other unbalanced roads from VSRivers for further investigation.

After she further explores the congestion conditions of the city with the system, she summarizes her findings with hypotheses on congestion causes generated by using our system. At first, several factors contribute to city's congestion. The first one is the river that horizontally divides the city. As rivers tend to generate large bridge-crossing traffic, both the quantity and locations of bridges need to be carefully decided as a city grows. In this point of view, a large volume of traffic on bridge-crossing had been expected between a university area (southern S1) and residential areas (northern S1), but only one bridge exists between them. Additionally, the congestion condition of S1 is exacerbated by a highway gate, as represented by the black traffic circle in S1.

One purpose of using a traffic circle is to allow vehicles to pass through the area faster than an intersection. But when its capacity is exceeded with a large traffic volume, it leads to congestion, which seemed to occur in this city. For example, two traffic circles exist in this city, one in S1 (black circle) and another in S2 (black circle), each of which had severe congestion. Specifically, a large volume is observed in the circle in S2, as vehicles heading for the central business districts (S2 east), industry complexes (below S2), and downtown (S3) use this circle. In this situation, it is estimated that the circle had been adversely affecting traffic in the city, exacerbating the congestion problem in S2. Industrial complexes also generate large logistical problems due to commuter traffic. This is the case in S4. Originally, the main traffic in S4 was vehicles moving north toward another city. After an industrial complex was developed above S4, new traffic was generated in S4. Even worse, with the new traffic, the bridge capacity problem also happened, causing severely congested roads in S4.

7.2 Investigation on Congestion Improvement Projects

In order to relieve the congestion problem, as revealed in the first study, the city has devoted considerable effort towards future urban planning. Congestion improvement projects are a part of this effort, where traffic signals and lane capacity are optimized.

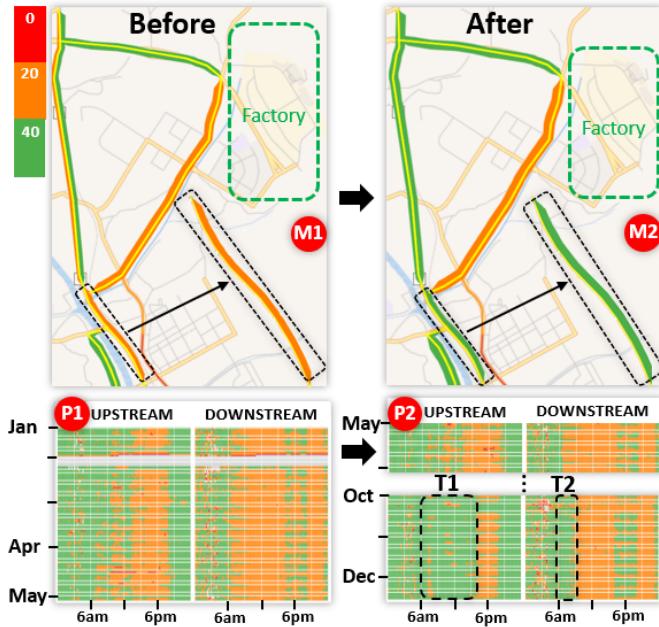


Fig. 10. Comparison of a congestion improvement project result. (M1) and (P1) present congestion conditions before the project, while (M2) and (P2) describe the result of the project. The orange road (magnified, M1) and pixels (P1) becomes green in M2 and P2.

Multiple steps are taken in the project over 6–8 months for surveying problematic roads, estimating project impact, performing the project, and measuring the resultant impact. In this case study, we demonstrate how our system can be used to investigate the impacts of a previous project and search for candidate roads.

Two main industrial complexes have been developed in the city. Since congestion around the complexes will adversely affect the complexes' success (e.g., increased logistics cost and time), the city has been concerned with the congestion. As such, the city performed a congestion improvement project on a main road strip of a complex from May 2017 to Oct 2017. The main strip road is magnified and presented within the black box in Fig. 10 (M1) and (M2). In order to determine the project's effectiveness, an analyst starts using VSRivers and quickly notices that the orange strip in (M1) was changed to a green strip in (M2) after the project. Then, DetectorVis is used to provide a detailed hourly analysis, as shown in Fig. 10 (P1 before project) and (P2 after project). Here the analyst observes in (P1) that congestion frequently happened during 6am–8pm in both streams. Specifically, the downstream congestion seems worse than that of the upstream, as she sees many orange pixels even after 8pm. The result of the project is presented in (P2). She sees that the upstream congestion became better (T1) with many green pixels (i.e., no congestion). However, the roads in the downstream still have many orange pixels except 6–8am and 4–8pm (T2). Based on these observations, the analyst concludes that the project was partially successful, and further investigation is needed to explore the ineffectiveness of the project on the downstream traffic.

The next task is to find a candidate road whose congestion conditions can be improved with the project. Conventionally, this task has been performed based on a field survey, where a few employees are dispatched to roads and count vehicles on the roads. This approach, however, is subject to high man-hour costs and a time-consuming data collection process. For example, in the study

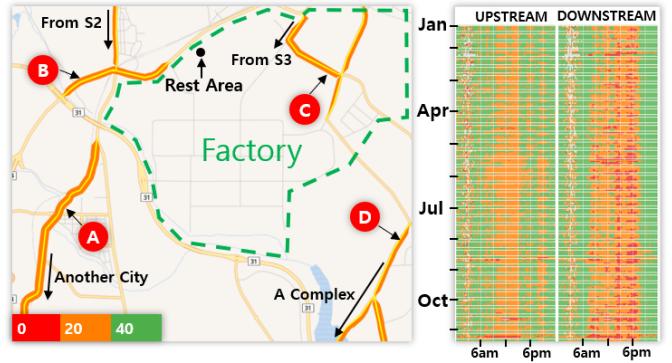


Fig. 11. Candidate roads (A–D) for a congestion improvement project are investigated with VSRivers (Left) and pixel visualization (Right).

described above, it took 4 months (Jan–Apr 2017) to collect data and select the main strip as the final target road. In this situation, the analyst uses the system to find a candidate road that meets three conditions: 1) frequently or severely congested; 2) near an industrial complex to help logistics flow, and; 3) connected to other roads to facilitate an additional positive impact. As conditions 2) and 3) are geographical criteria, the exploration is mainly performed with VSRivers and zooming and panning.

An example candidate complex is presented in Fig. 11 (left), where geographical conditions are shown (left). Here, the analyst sees several congested roads (e.g., A–D) with VSRivers, each of which seems to have a similar traffic volume and congestion. There are factors that make these roads congested. For example, B and C are congested due to high volumes coming from Fig. 5 S2 and S3. The volume on A is mainly for going to another city, while that of D is from C and heading to another complex. All these roads can be a candidate, but in this case, the analyst thinks that B has the highest priority, since B is located at the entrance of the complex and has the largest number of connected roads. Road C can also be considered, but it is located inside the complex, which implies few factories would benefit from the project.

7.3 Broadcasting Traffic Congestion Conditions

While our previous case studies demonstrate how our system can be used for planning and retrospective analysis, the other key feature of our work is the real-time monitoring and prediction. Reporters in a radio station broadcast congestion conditions eight times per hour during peak commute times (7am–9am, 6pm–8pm). Otherwise they report conditions four times per hour. Each broadcast must be finished within 3 minutes. Thus, the main task of the reporters is to prepare a script that best describes current and future congestion conditions on important roads. However, the script preparation is not easy. Typically, the reporter has only 5 minutes to both understand congestion conditions across the city and make decisions on which roads to broadcast with congestion prevention advice for drivers (e.g., detours). To assist the preparation process, two staff members are assigned to record important events (e.g., accidents) and congestion conditions and produce summary notes while watching 128 CCTVs and receiving phone calls from volunteers. However, there is always a chance that the staff members will miss critical events or congestion on roads due to wrong CCTV angles and an insufficient number of calls from volunteers.

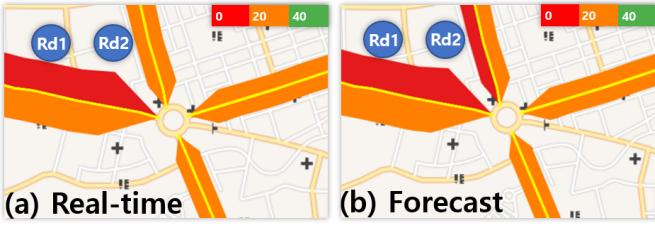


Fig. 12. An example comparison of real-time (a) and forecasted (b) congestion conditions around a traffic circle. The forecasting result (b) indicates that Rd2 will be severely congested in 15 minutes.

To improve script preparation, a reporter uses our system in the real-time mode that updates congestion conditions every minute. At first, she looks at VSRivers, comparing the congestion conditions from VSRivers to the notes produced by the staff members. For example, she compares congestion conditions of S1–S4 in Fig. 5 (D) to the provided notes to confirm her domain knowledge on frequently congested areas. Next, she turns on the CCTV videos in the system to correct any mismatched, incorrect, out-of-date, or invalid information on the report. Once the information from the two data sources does not have any issues, she uses the road list view, as shown in Fig. 5 (B) and sorts the list such that highly congested roads are shown from the top with estimated travel times (ETTs). She notes the most congested roads and their ETTs. Once every step up to this point does not have any issues, she selects roads in the map view to be broadcast and places the roads in both the CCTV view and the snapshot view (Fig. 5 F) for real-time monitoring during broadcasting. If there is any newly released information at this point, she updates her script with current information.

For the last step of preparing congestion avoidance advice for drivers, she turns on the forecasting mode where she can develop an idea of how current congestion conditions may evolve in the next 15 minutes. An example is shown in Fig. 12, where VSRivers present both real-time (a) and forecasted (b) congestion conditions around the traffic circle in Fig. 5 S2 at 8:15am. Here she finds that Rd1 in Fig. 12 (a) is the only impeded road. By looking at the forecasted congestion conditions in (b), she understands that Rd2 is soon to be impeded. To double-check the plausibility of this result, she turns on the PropArrows visualization, which is PropArrow1 in Fig. 7. Note that P2 in Fig. 7 is the same traffic circle in Fig. 12. From PropArrow1, she quickly notices a possible propagation pattern that 1) P1–P2 first became congested (i.e., a possible cause), followed by P2–P3 and 2) this congestion could last for 15 minutes. With the given information and insights, she looks at the ETT values of the congested roads around the traffic circle in the table view, as shown in Fig. 7 bottom-left, and notices that passing P3–P2–P1 requires at least 6 minutes longer than driving without congestion. Based on the information, she advises that detouring the traffic circle at this time is suggested.

8 EXPERT FEEDBACK

Three domain experts (i.e., the group leader from the city congestion management center and the assistant director and the reporter from the radio broadcasting center) participated an 60-minute informal interview session and provided positive feedback on the system. Before the session, the experts used our system for 20–30 minutes with a short training session on how to use our tool. In the interview session, we asked experts which of their

daily tasks were most aided by the system and how their task processes can be changed with our system. We asked them to describe their tasks with respect to the design requirements we elicited from previous interviews with this cohort. The assistant director from the radio center said that the main difficulty in congestion reporting is a lack of tools that can provide summaries of congestion conditions across the city in real-time. This lack of effective tools at the center has caused many difficulties in reporting; “...it is very hard to analyze congestion conditions of all of the roads in the city within 5 minutes,” reported one expert. As such, well-known congested areas tend to get all of the focus during her reporting (e.g., S1–S4 in Fig. 5 A), while there could be urgent situations on other roads due to unexpected reasons (e.g., accidents). Additionally, they have only considered roads with CCTVs in their reporting, as congestion on roads without CCTVs cannot be monitored and analyzed. CCTVs have not yet been installed on the roads around a newly built industrial complex, despite its importance. “...the system is very useful, as the visualization in the system allows quick overviews of congestion conditions of roads including the road without CCTVs in real-time,” the assistant director mentioned. The reporter also commented that using the VSRivers, forecasting, ETT values, and the PropArrows visualization functionality of the system can be very helpful as it not only provides information in real-time but also enhances confidence during reporting. For example, reporters have intentionally used vague words and sentences in reporting, such as “vehicles on the road are stop-and-go.” There have been complaints from listeners because such vagueness does not give any sense of how slow the traffic is at drive time. With forecasted congestion, estimated travel time, and speed information, the sentences can be changed to “...vehicles on the road are 5 km/hour slower than usual, so driving this road may take at least 6 minutes longer than usual,” the reporter clarified.

The group leader from the city congestion management center reported positive feedback. The leader said VSRivers with the calendar and clock visualizations can be very useful in performing their task of analyzing congestion conditions with historical data. Performing the analytics task is important in that results of such analytics are direct inputs of the congestion improvement projects and the city’s official reports which analyze congestion problems and discuss possible solutions. Specifically, he easily recognized the usefulness of the DetectorVis and said that “...when combined with VSRivers, it (i.e., DetectorVis) is beneficial in not only searching problematic roads for city congestion improvement projects but also investigating past project results.” He reported another benefit of the system is that DetectorVis can significantly reduce the burden of searching for erroneous vehicle detectors that should be replaced.

9 LIMITATIONS AND DISCUSSION

There are studies that use auxiliary traffic information to improve forecasting accuracy, such as weather information [59]. But, we have not seen any increased performance with the information in our experiments. One reason might be that auxiliary information is not a critical factor within complex road networks. For example, compared to previous studies with relatively simple road networks such as rings, belts, or long linear freeways with very few connected roads [36], [40], we use more complex data where each road is linked to 4 other roads on average (max: 21 linked roads). Adding traffic flow as a new feature does not make big difference

in the performance result, as the speed feature already reflects the flow characteristics according to the speedflow relationship [60]. Another possibility is that the model we used might not be the best one without structural changes to best utilize the auxiliary information. As such, more experiments with different types of prediction architectures should be considered. Lastly we have not considered flow information, because flow fluctuation is reflected to speed–vehicle speeds drop as flow increases, as shown in other flow-speed relationship studies.

Retraining prediction models can be beneficial, as congestion patterns can change over time. But it is hard to conclude the best model re-training frequency without empirical studies, because more retraining means increased costs. If three-month data can lead to the best performance in terms of accuracy, users can train the model every three months at a relatively low cost (e.g., 133 minutes in this work). When a longer training time is required, other approaches such as online learning [61] can be considered; here, a model can be trained as new data come in. We leave the empirical studies on finding the best model re-training timing and frequency as future work. It is also important to note that all models have errors. Forecasting errors could lead to erroneous traffic reports, and future iterations of the system will explore mechanisms to display uncertainty surrounding the forecasts [62].

10 CONCLUSION AND FUTURE WORK

We have introduced a visual analytics system for exploring, monitoring, and forecasting traffic congestion on roads, as a result of collaboration with domain experts over 18 months. Our system provides experts with several advantages. First, our system enables domain experts to have overviews of city-wide congestion conditions by utilizing multiple-coordinate views and allowing many filtering options, such as dates, time, speed and volume. Specifically, VSRivers visualization allows efficient congestion analyses by simultaneously providing volume and congestion conditions. To help users, whose mission is understanding and forecasting congestion conditions, causes and propagation directions, we design a weighting method, PropArrows visualization, and incorporate a deep learning model. Finally, we quantitatively and qualitatively evaluate our system by comparing forecasting algorithm performance and present 3 case studies including interviews with domain experts. In future work, we plan to perform more rigorous experiments by forecasting with other factors (e.g., weather, accidents) to develop more accurate forecasting models.

REFERENCES

- [1] J. Zhang, F. Y. Wang, K. Wang, W. H. Lin, X. Xu, and C. Chen, “Data-driven intelligent transportation systems: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.
- [2] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan, “Tripvista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection,” in *Proceedings of IEEE Symposium on Pacific Visualization*, 2011, pp. 163–170.
- [3] H. Piringer, M. Buchetics, and R. Benedik, “Alvis: Situation awareness in the surveillance of road tunnels,” in *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, 2012, pp. 153–162.
- [4] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. van de Wetering, “Visual traffic jam analysis based on trajectory data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2159–2168, 2013.
- [5] Z. Wang, T. Ye, M. Lu, X. Yuan, H. Qu, J. Yuan, and Q. Wu, “Visual exploration of sparse traffic trajectory data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1813–1822, 2014.
- [6] F. Wang, W. Chen, F. Wu, Y. Zhao, H. Hong, T. Gu, L. Wang, R. Liang, and H. Bao, “A visual reasoning approach for data-driven transport assessment on urban roads,” in *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, 2014, pp. 103–112.
- [7] J. C. Roberts, “State of the art: Coordinated multiple views in exploratory visualization,” in *Proceedings of International Conference on Coordinated and Multiple Views in Exploratory Visualization*, 2007, pp. 61–71.
- [8] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [9] D. A. Keim, “Designing pixel-oriented visualization techniques: theory and applications,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 59–78, 2000.
- [10] Y.-C. Chen, Y.-S. Wang, W.-C. Lin, W.-X. Huang, and I.-C. Lin, “Interactive visual analysis for vehicle detector data,” *Computer Graphics Forum*, vol. 34, no. 3, pp. 171–180, 2015.
- [11] G. N. Oliveira, J. L. Sotomayor, R. P. Torchelsen, C. T. Silva, and J. L. D. Comba, “Visual analysis of bike-sharing systems,” *Computers & Graphics*, vol. 60, pp. 119–129, 2016.
- [12] J. Pu, S. Liu, Y. Ding, H. Qu, and L. M. Ni, “T-watcher: A new visual analytic system for effective traffic surveillance,” in *Proceedings of IEEE Conference on Mobile Data Management*, vol. 1, 2013, pp. 127–136.
- [13] G. Sun, R. Liang, H. Qu, and Y. Wu, “Embedding spatio-temporal information into maps by route-zooming,” *IEEE transactions on visualization and computer graphics*, vol. 23, no. 5, pp. 1506–1519, 2017.
- [14] W. C. 0001, Z. Huang, F. Wu, M. Zhu, H. Guan, and R. Maciejewski, “Vaud: A visual analysis approach for exploring spatio-temporal urban data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 9, pp. 2636–2648, 2018.
- [15] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva, “Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2149–2158, 2013.
- [16] S. Liu, J. Pu, Q. Luo, H. Qu, L. M. Ni, and R. Krishnan, “Vait: A visual analytics system for metropolitan transportation.” *IEEE Transactions Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1586–1596, 2013.
- [17] W. Zeng, C.-W. Fu, S. M. Arisona, and H. Qu, “Visualizing interchange patterns in massive movement data,” *Computer Graphics Forum*, vol. 32, no. 3, pp. 271–280, 2013.
- [18] W. Zeng, C.-W. Fu, S. M. Arisona, A. Erath, and H. Qu, “Visualizing mobility of public transportation system,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1833–1842, 2014.
- [19] L. Yu, W. Wu, X. Li, G. Li, W. S. Ng, S.-K. Ng, Z. Huang, A. Arunan, and H. M. Watt, “iviztrans: Interactive visual learning for home and work place detection from massive public transportation data,” in *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, 2015, pp. 49–56.
- [20] G. D. Lorenzo, M. L. Sbodio, F. Calabrese, M. Berlingero, F. Pinelli, and R. Nair, “Allaboard: Visual exploration of cellphone mobility data to optimise public transport,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 2, pp. 1036–1050, 2016.
- [21] C. Palomo, Z. Guo, C. T. Silva, and J. Freire, “Visually exploring transportation schedules,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 170–179, 2016.
- [22] W. Chen, F. Guo, and F.-Y. Wang, “A survey of traffic data visualization,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2970–2984, 2015.
- [23] Y. Zheng, W. Wu, Y. Chen, H. Qu, and L. M. Ni, “Visual analytics in urban computing: An overview.” *IEEE Transactions on Big Data*, vol. 2, no. 3, pp. 276–296, 2016.
- [24] G. Andrienko, N. Andrienko, W. Chen, R. Maciejewski, and Y. Zhao, “Visual analytics of mobility and transportation: State of the art and further research directions,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2232–2249, 2017.
- [25] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, “Short-term traffic forecasting: Where we are and where were going,” *Transportation Research Part C: Emerging Technologies*, vol. 43, no. Part 1, pp. 3 – 19, 2014.
- [26] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, “Short-term traffic forecasting: Overview of objectives and methods,” *Transport reviews*, vol. 24, no. 5, pp. 533–557, 2004.
- [27] M. Levin and Y.-D. Tsao, “On forecasting freeway occupancies and volumes,” *Transportation Research Record*, no. 773, pp. 47–49, 1980.
- [28] Y. Zhang and Y. Liu, “Traffic forecasting using least squares support vector machines,” *Transportmetrica*, vol. 5, no. 3, pp. 193–213, 2009.
- [29] M. Karlaftis and E. Vlahogianni, “Statistical methods versus neural networks in transportation research: Differences, similarities and some

- insights,” *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387 – 399, 2011.
- [30] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [31] W. Huang, G. Song, H. Hong, and K. Xie, “Deep architecture for traffic flow prediction: Deep belief networks with multitask learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.
- [32] Y. Lv, 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, pp. 865–873, 2015.
- [33] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, “Long short-term memory neural network for traffic speed prediction using remote microwave sensor data,” *Transportation Research Part C: Emerging Technologies*, vol. 54, no. Supplement C, pp. 187 – 197, 2015.
- [34] Y. Wu and H. Tan, “Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework,” *arXiv preprint arXiv:1612.01022*, 2016.
- [35] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, “Spatiotemporal recurrent convolutional networks in transportation networks,” *arXiv preprint arXiv:1705.02699*, 2017.
- [36] Z. Cui, R. Ke, and Y. Wang, “Deep stacked bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction,” in *Proceedings of ACM SIGKDD International Workshop on Knowledge Discovery and Data Mining*, 2017.
- [37] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, “Lstm network: a deep learning approach for short-term traffic forecast,” *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.
- [38] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Graph convolutional recurrent neural network: Data-driven traffic forecasting,” *arXiv preprint arXiv:1707.01926*, 2017.
- [39] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, “Deep learning: A generic approach for extreme condition traffic forecasting,” in *In Proceedings of SIAM International Conference on Data Mining*, 2017, pp. 777–785.
- [40] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting,” *arXiv preprint arXiv:1709.04875*, 2017.
- [41] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” *arXiv preprint arXiv:1312.6203*, 2013.
- [42] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.
- [43] J. B. Kenney, “Dedicated short-range communications (dsrc) standards in the united states,” *Proceedings of IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.
- [44] J. Gajda, R. Sroka, M. Stencel, A. Wajda, and T. Zeglen, “A vehicle classification based on inductive loop detectors,” in *Proceedings of IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics*, 2001, pp. 460–464.
- [45] *Highway capacity manual*, 113th ed., Transportation research board, National Research Council, 2000.
- [46] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Mueller, “Efficient BackProp,” *Lecture Notes in Computer Science*, vol. 1524, pp. 9–50, 1998.
- [47] P. Goyal and E. Ferrara, “Graph embedding techniques, applications, and performance: A survey,” *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.
- [48] H. Cai, V. W. Zheng, and K. C.-C. Chang, “A comprehensive survey of graph embedding: Problems, techniques and applications,” *Computing Research Repository*, vol. abs/1709.07604, 2017.
- [49] S. Mannor, D. Peleg, and R. Rubinstein, “The cross entropy method for classification,” in *Proceedings of the international conference on Machine learning*, 2005, pp. 561–568.
- [50] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [52] R. Caruana, S. Lawrence, and C. L. Giles, “Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping,” in *Advances in neural information processing systems*, 2001, pp. 402–408.
- [53] “W. beaty. traffic waves, sometimes one driver can vastly improve traffic,” <http://trafficwaves.org/>, 1998.
- [54] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, “Traffic jams without bottleneck: experimental evidence for the physical mechanism of the formation of a jam,” *New journal of physics*, vol. 10, no. 3, p. 033001, 2008.
- [55] M. D. Plumlee and C. Ware, “Zooming versus multiple window interfaces: Cognitive costs of visual comparisons,” *ACM Computer Human Interaction*, vol. 13, no. 2, pp. 179–209, 2006.
- [56] D. Sacha, F. Al-Masoudi, M. Stein, T. Schreck, D. A. Keim, G. Andrienko, and H. Janetzko, “Dynamic visual abstraction of soccer movement,” *Computer Graphics Forum*, vol. 36, no. 3, pp. 305–315, 2017.
- [57] S. R. Chandra and H. Al-Deek, “Predictions of freeway traffic speeds and volumes using vector autoregressive models,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 53–72, 2009.
- [58] A. Ding, X. Zhao, and L. Jiao, “Traffic flow time series prediction based on statistics learning theory,” in *Proceedings of IEEE International Conference on Intelligent Transportation Systems*, 2002, pp. 727–730.
- [59] Y. Jia, J. Wu, M. Ben-Akiva, R. Seshadri, and Y. Du, “Rainfall-integrated traffic speed prediction using deep learning method,” *IET Intelligent Transport Systems*, vol. 11, no. 9, pp. 531–536, 2017.
- [60] B. L. S. Hemant Kumar Sharma, Mansha Swami, “Speed-flow analysis for interrupted oversaturated traffic flow with heterogeneous structure for urban roads,” *International Journal for Traffic and Transport Engineering*, vol. 2, no. 2, pp. 142–152, 2012.
- [61] J. L. P. Z. Steven C.H. Hoi, Doyen Sahoo, “Online learning: A comprehensive survey,” *arXiv preprint arXiv:1802.02871*, 2018.
- [62] D. Sacha, H. Senaratne, B. C. Kwon, G. P. Ellis, and D. A. Keim, “The role of uncertainty, awareness, and trust in visual analytics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 240–249, 2016.



Chunggi Lee received his B.S. in school of electrical and computer engineering from UNIST (Ulsan National Institute of Science and Technology), South Korea. He is currently an MS student in the school of computer engineering in UNIST. His research interests include visual analytics, HCI, and Deep Learning.



Yeonjun Kim is an undergraduate student in the school of engineering at UNIST (Ulsan National Institute of Science and Technology), South Korea. His research interests include visual analytics and HCI.



Seungmin Jin is a researcher of electrical and computer engineering at UNIST (Ulsan National University of Science and Technology). His research interests include visual analytics, statistical analysis, and Explainable AI (XAI). He acquired MSc in Big data systems from Higher School of Economics, Moscow, Russia in 2018. Prior to joining UNIST, he was a Big data researcher at Daumsoft Mining Lab.



Dongmin Kim is an undergraduate student in the school of electrical and computer engineering at UNIST (Ulsan National Institute of Science and Technology), South Korea. His research interests include visual analytics, HCI and virtual reality.



Ross Maciejewski is an Associate Professor at Arizona State University in the School of Computing, Informatics & Decision Systems Engineering. His primary research interests are in the areas of geographical visualization and visual analytics focusing on public health, dietary analysis, social media, criminal incident reports, and the food-energy-water nexus. For more information, visit <http://vader.lab.asu.edu>.



David Ebert is the Silicon Valley Professor of Electrical and Computer Engineering at Purdue University, a Fellow of the IEEE, and Director of the Visual Analytics for Command Control and Interoperability Center (VACCINE), the Visualization Science team of the Department of Homeland Security's Command Control and Interoperability Center Emeritus of Excellence. Dr. Ebert performs research in novel visualization techniques, data and visual analytics, volume rendering, information visualization, perceptually-based visualization, illustrative visualization, and procedural abstraction of complex, massive data.



Sungahn Ko is an assistant professor in the school of electrical and computer engineering at UNIST (Ulsan National Institute of Science and Technology), South Korea. His research interests include visual analytics, information visualization, and Human-Computer Interaction. He received the doctoral degree in electrical and computer engineering from Purdue University in 2014. For more information, visit <http://ivader.unist.ac.kr>