DGeye: Probabilistic Risk Perception and Prediction for Urban Dangerous Goods Management

JINGYUAN WANG and XIN LIN, School of Computer Science and Engineering, Beihang University. Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University. State Key Laboratory of Software Development Environment, Beihang University
YUAN ZUO and JUNJIE WU, School of Economics and Management, Beihang University. Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University. Beijing Key Laboratory of Emergency Support Simulation Technologies for City Operations, Beihang University

Recent years have witnessed the emergence of worldwide megalopolises and the accompanying public safety events, making urban safety a top priority in modern urban management. Among various threats, dangerous goods such as gas and hazardous chemicals transported through cities have bred repeated tragedies and become the deadly "bomb" we sleep with every day. While tremendous research efforts have been devoted to dealing with dangerous goods transportation (DGT) issues, further study is still in great need to quantify this problem and explore its intrinsic dynamics from a big data perspective. In this article, we present a novel system called DGeye, to feature a fusion between DGT trajectory data and residential population data for dangers perception and prediction. Specifically, DGeye first develops a probabilistic graphical model-based approach to mine spatio-temporally adjacent risk patterns from population-aware risk trajectories. Then, DGeye builds the novel causality network among risk patterns for risk pain-point identification, risk source attribution, and online risky state prediction. Experiments on both Beijing and Tianjin cities demonstrate the effectiveness of DGeye in real-life DGT risk management. As a case in point, our report powered by DGeye successfully drove the government to lay down gas pipelines for the famous Guijie food street in Beijing.

CCS Concepts: • **Information systems → Information systems applications**; **Spatial-temporal systems**; **Data mining**;

Additional Key Words and Phrases: Urban safety, dangerous goods transportation, risk management, risk pattern, risk causal network

This work was supported by the National Key R&D Program of China (2019YFB2101804). Dr. Jingyuan Wang's work was partially supported by the National Natural Science Foundation of China (92046010, 82161148011), the Fundamental Research Funds for the Central Universities (YWF-21-BJ-J-839), and CCF-DiDi Gaia Collaborative Research Funds for Young Scholars. Dr. Junjie Wu's work was partially supported by the National Natural Science Foundation of China (71725002, 71531001, 72031001, 72021001). Dr. Yuan Zuo's work was partially supported by the National Natural Science Foundation of China (71901012).

Authors' addresses: J. Wang and X. Lin, School of Computer Science and Engineering, Beihang University. Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University. State Key Laboratory of Software Development Environment, Beihang University, Beijing, China; emails: {jywang, sweeneylin}@buaa.edu.cn; Y. Zuo and J. Wu (corresponding author), School of Economics and Management, Beihang University. Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University. Beijing Key Laboratory of Emergency Support Simulation Technologies for City Operations, Beihang University, Beijing, China; emails: {zuoyuan, wujj}@buaa.edu.cn.

1046-8188/2021/05-ART28 $15.00
https://doi.org/10.1145/3448256

ACM Reference format:
Jingyuan Wang, Xin Lin, Yuan Zuo, and Junjie Wu. 2021. DGeye: Probabilistic Risk Perception and Prediction for Urban Dangerous Goods Management. *ACM Trans. Inf. Syst.* 39, 3, Article 28 (May 2021), 30 pages.
https://doi.org/10.1145/3448256

1 INTRODUCTION

Dangerous goods refer to materials that are radioactive, flammable, explosive, or toxic. Rapid industrialization associated with urbanization leads to increasing dangerous goods applications in densely populated urban areas and the subsequent massive transport and storage of dangerous goods, which have exposed huge risks to modern cities. For instance, a warehouse storing dangerous goods in the port area of Tianjin, exploded on August 12, 2015, causing 173 deaths, hundreds of wounded, and 304 buildings severely damaged.[1] On August 4, 2020, a massive explosion at the port of the city of Beirut, the capital of Lebanon, caused at least 180 deaths, 6,000 injuries, and US$10–15 billion in property damage, leaving an estimated 300,000 people homeless.[2] These bloody lessons bring urban dangerous goods safety back to sight as a top priority in urban management. Discovering and managing the risks caused by dangerous goods have become an essential task for urban governments and a big challenge to researchers involved.

In the literature, the problem of **dangerous goods transportation (DGT)** has attracted great attention, but mainly focused on transportation route planning [24] and risk assessment [14] and mostly from an operation and optimization view. These studies, though providing constructive managerial insights, usually lack a micro-view of DGT threats from a big data perspective, which is deemed critically important for gaining actionable rules. For instance, we need to learn the geographical distributions of dangerous goods transporters (DGT)[3] and urban residents for real-time risk monitoring. Also, we should identify the latent spatio-temporal risk patterns and the intrinsic mechanism behind them for risk perception and prediction as well as sustainable urban planning. These practical needs indeed motivate our study in this article, which aims to leverage heterogeneous big data to deal with urban risks stemming from DGT. Our study falls into the research category of urban computing [43] and thus can also enrich the dangerous goods-related studies in this area. The key research question in this work is how to identify the latent spatio-temporal risk patterns and the intrinsic mechanism behind them from dangerous-goods-related big data. The key question could be divided into three specific research questions:

(1) How to identify spatio-temporal risk patterns from DGT trajectory and human mobility data.
(2) How to model intrinsic relations among the spatio-temporal risk patterns and exploit the relations for urban risk management.
(3) How to predict urban DGT risks based on spatio-temporal risk pattern mining and pattern relations modeling.

In this work, we present a system called ***City Eyes on Dangerous Goods*** (DGeye) for real-world DGT safety management to answer the three research questions. DGeye takes spatio-temporal *risk patterns* as the "magic wand" for urban DGT risk analysis and adopts three probabilistic-approach-based components for risk pattern mining, risk sources tracking, and risky states prediction. In the "risk pattern mining" component, DGeye designs a novel probabilistic graphical model, named

[1] https://en.wikipedia.org/wiki/2015_Tianjin_explosions.
[2] https://en.wikipedia.org/wiki/2020_Beirut_explosion.
[3] We use DGT to denote both Dangerous Goods Transportation and Transporter interchangeably, which can be distinguished with reference to the context.

nRTG, to mine urban risk patterns from a joint dataset consisting of DGT trajectory data and resident mobile-phone signaling data. The nRTG model is also equipped with a neighbor regularization to model the spatio-temporal connections in urban risk patterns. In the "risk source tracking" component, a novel trajectory-driven *risk causal network* is built upon found risk patterns for pattern importance ranking and risk attribution analysis. In the "risky state prediction" component, DGeye is capable of predicting risk patterns' risky states by applying an EM-enabled Bayesian network model to the generated causal network. DGeye has established itself as a successful deployment in various real-world applications. For instance, as a look-back to the Tianjin port explosion disaster, DGeye accurately located the explosion site as one urban zone inside the top-ranked risk pattern in the port area. More interestingly, DGeye disclosed that the first and most important risk source in Beijing is formed by the transportation of liquefied gas cylinders to an old famous food street: Guijie.[4] The advisory report technically powered by DGeye has driven the Beijing government to lay down gas pipelines for Guijie.

Our main research contributions to DGeye are summarized as follows: First, to our best knowledge, DGeye (and its previous version [30]) is among the earliest data-driven systems that take both vehicle trajectories and human mobilities into account for DGT safety management and gain successful deployment. Second, DGeye is among the very few works that apply probabilistic graphical model to mine spatio-temporal patterns from two dynamic data sources simultaneously. Compared with our previous work in Reference [30], which adopts an Apriori-like algorithm to discover spatial-only risk patterns, the nRTG model proposed in this work can uncover spatio-temporally adjacent risk patterns and has the potential to suit other spatio-temporal pattern mining scenarios such as co-location pattern mining. Third, in DGeye, we design a trajectory-driven risk causal network to describe influential relationships among risk patterns, based on which a risk source tracking algorithm is proposed to rank the priority of risk patterns for treatment allocation. As shown in the experiments on the Beijing and Tianjin datasets, following the priority ranks generated by our approach, the decrease of the holistic risk of all patterns can reach 50% even if only the top-five patterns are treated. Fourth, we propose an EM-enabled Bayesian network model in DGeye to predict risky states of patterns consecutively. Comparative experimental studies with various baselines demonstrate the excellent predictive power of DGeye, especially when we have only a small data sample. Moreover, compared with our previous work in Reference [30], the new version of the DGeye system also achieved improved performance over the Beijing and Tianjin real-world datasets for both the risk source tracking and pattern risk state prediction applications.

The remainder of this article is organized as follows: Section 2 briefly introduces the DGeye system. Section 3 present the data precessing step of the DGeye system. Sections 4 to 6 present the technical details of the three functional components of DGeye orderly. We present the experimental results in Section 7, introduce related work in Section 8, and conclude our work in Section 9.

2 THE SYSTEM OVERVIEW

Figure 1 shows the framework of DGeye with four layers. The data source layer of the system consists of DGT trajectory data, mobile phone signaling data, and city map data, which, respectively, represent the information about dangerous goods, human populations, and city geography. In the data processing layer, the system partitions a city map into multiple urban zones and spatio-temporal zones, and then uses mobile phone signaling data and DGT trajectory data to calculate the crowd scores for each urban zones and risk trajectories (see Section 3 for details).

[4] https://www.youtube.com/watch?v=gP08plZygHE.

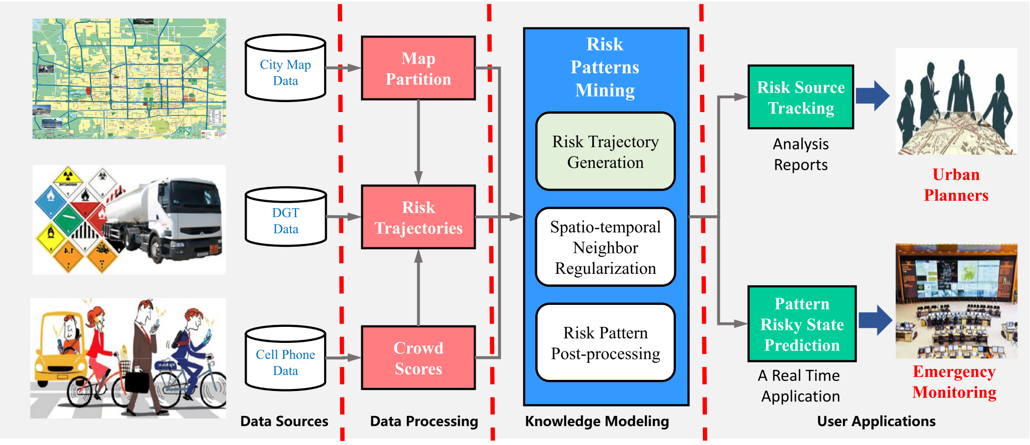


Fig. 1. The system framework of DGEYE.

The knowledge modeling layer is concerned with the pain points of DGT risks within a city. While crowd scores in urban zones and risk trajectories are important for real-time monitoring, they are just the "irregular symptoms" of the underlying DGT risks, changing across different time slices and in different days. For an urban management perspective, we would like to unveil the relatively stable patterns behind the time-variable symptoms. Therefore, we adopt a *Risk Pattern Mining* component as a knowledge modeling layer, which reveals the spatio-temporal patterns of risks from crowd scores and risk trajectories (see Section 4 for details). The DGEYE system implements this function via a ***neighbor-regularized Risk Trajectory Generation* (nRTG)** model, which is a probabilistic graphical model describing the generation process of DGT trajectories and a spatio-temporal neighbor regularization mechanism. A pattern postprocesses step is also adopted to ensure risk patterns satisfy frequency, connectivity, and crowdedness requirements.

Based on the risk patterns, in the user application layer, we develop two applications for different types of users. For urban planners, the system adopts a *Risk Source Tracking* component to pick out the risk patterns (i.e., risk sources) that have high influences on the formation of other risk patterns (see Section 5 for details). To this end, the risk patterns generated previously are ranked according to their causal influences to other patterns. The ranking gives high priority to the patterns that lead to many other patterns of high importance. According to the ranking list, urban planners can fix the pain points gradually from high-priority patterns to low-priority ones. The system implements this function through building a causal network among risk patterns and running a probabilistic random walk algorithm over the network. For the emergency monitoring application, the system adopts a *Risky State Prediction* component to predict the risky states of risk patterns in the next time slice based on their historical states and the mutual influences defined in the causal network (see Section 6 for details). The DGEYE system adopts an EM-enabled Bayesian approach over the risk pattern causal network to fulfill this task.

3 DATA PROCESSING

The input of the system mainly consists of three types of data sources, i.e., the map of a city, the DGT trajectory, and the mobile phone signaling data. In this section, we introduce the data sources and processing step of the data sources in DGEYE.

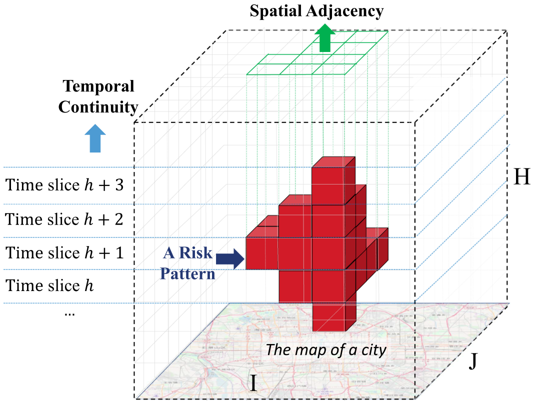


Fig. 2. An illustration of the ST-Cube and risk patterns.

3.1 Spatio-Temporal Zone

In the DGeye system, the map of a city is segmented as many basic units, namely, *Urban Zones* and *ST-Zones*, for DGT risk analyzing.

Definition 1 (Urban Zone). We divide the map of a city as a $I \times J$ checkerboard. A grid in a checkerboard is defined as an *Urban Zone*.

Definition 2 (ST-Zone). We divide one day into H time slices. A *Spatio-Temporal Zone*, abbreviated as *ST-Zone*, is defined as an instance of an urban zone for a time slice. Given a city with $I \times J$ urban zones, there are $V = I \times J \times H$ ST-Zones. The ST-Zone corresponding to the (i, j)-th urban zone at the time slice h is denoted as s_{ijh}. We also use s_v to denote the vth ST-Zone if all ST-Zones are lined up as a vector, $v = 1, \ldots, V$.

Definition 3 (ST-Cube). All ST-Zones constitute a *Spatio-Temporal Cube*, abbreviated as *ST-Cube*, with the size of $I \times J \times H$ (as shown in Figure 2). The ST-Zone s_{ijh} is the (i, j, h)-th cell of the ST-Cube.

3.2 Crowd Score

For each ST-Zone, the DGeye system generates a *Crowd Score* using *Mobile Phone Signaling (MPS) Data*. The MPS data refers to communication records between mobile phones and base stations. An MPS record contains <user ID, station ID, timestamp> fields. The user ID and base station ID are unique identifications for cell phones and base stations. The timestamp records the occurrence time of communication. The DGeye system uses the location of the base station that provides signaling services to a mobile phone to approximate the position of the phone user. Given the pretty high penetration rate of mobile phones in metropolises, we can use the number of mobile phone users to approximate the population in an urban zone. In our system, the user IDs are anonymous and the MPS data do not contain any communication contents of cell phone users.

For each time slice, the system maintains a binary user-zone matrix $\mathbf{U}$, where a row vector corresponds to a cell phone user and a column corresponds to a urban zone. The element $\tilde{u}_{xy} = 1$ indicates user x appears in urban zone y during the time slice, and 0 otherwise. At the end of a time slice, the population of the urban zone y is calculated as

$$u_y = \sum_x \frac{\tilde{u}_{xy}}{\sum_y \tilde{u}_{xy}}. \tag{1}$$

In this way, if a user visits K urban zones in a time slice, then the user is only counted $1/K$ times for each zone. Using the population of urban zones in each time slice, the DGeye system calculates *Crowd Score* of each ST-Zone.

Definition 4 (Crowd Score). Let u_{ijh} denote the population in the ST-Zone s_{ijh}, i.e., the population in the (i, j)-th urban zone at the hth time slice of a certain day. The *Crowd Score* of s_{ijh} in the day is then defined as

$$c_{ijh} = \sigma\left(\frac{u_{ijh} - \bar{u}}{\mathrm{std}(u)}\right), \tag{2}$$

where $\bar{u}$ and $\mathrm{std}(u)$ are, respectively, the mean and standard deviation of the populations in all ST-Zones and in all days, and the sigmoid function $\sigma(\cdot)$ maps c_{ijh} to the range of $(0, 1)$.

We use crowd scores to measure the relative population density level of an ST-Zone. Or equivalently, we can take the crowd score c_{ijh} as an associated population property of an ST-Zone s_{ijh}.

3.3 Risk Trajectory

DGT trajectories are collected by GPS terminals that are equipped with dangerous goods transporters. A DGT trajectory record contains < vehicle ID, location, speed, timestamp > fields, where the Vehicle ID field is a unique identification of a DGT, the location and speed fields record the real-time location and speed of a transporter, and the timestamp field records the report time of the record. From the DGT trajectory dataset, the DGeye system extracts *Risk Trajectories* for further risk analysis.

Definition 5 (Risk Trajectory). A *Risk Trajectory* is a set of ordered ST-Zones visited in sequence by a dangerous goods transporter in one transportation task. A risk trajectory with the index m is defined as $t_m = \{(z_{m1}, c_{m1}), \ldots, (z_{mn}, c_{mn}), \ldots, (z_{mN}, c_{mN})\}$, where $z_{mn} \in \{s_1, \ldots, s_V\}$ indicates the ST-Zone at the nth point of the trajectory t_m, with z_{mn} and $z_{m(n+1)}$ being two spatio-temporally adjacent ST-Zones, and c_{mn} is the associated crowd score of the ST-Zone z_{mn}.

For a raw trajectory of a DGT, if its z_{mn} and $z_{m(n+1)}$ are spatio-temporally adjacent, then we include the ST-Zones on the shortest path between z_{mn} and $z_{m(n+1)}$ into the risk trajectory. A risk trajectory contains not only the spatio-temporal location information of a dangerous goods transporter but also the population density at each location that can help indicate the potential risk.

4 RISK PATTERN MINING

In this section, we define the key concept of *Risk Pattern* and propose a probabilistic framework for risk pattern mining. Table 1 gives the math notations.

Definition 6 (Risk Pattern). A *Risk Pattern* is a set of spatio-temporally adjacent ST-Zones that are fully connected in a ST-Cube (see Figure 2), frequently visited by dangerous goods transporters, and crowded with relatively dense populations.

A risk pattern can be regarded as an *episode* of risk trajectories frequently adopted by DGT drivers in their route planning and passing by crowded city areas. Hence, the *frequency*, *connectivity* and *crowdedness* are three key features that define a risk pattern. In what follows, we will give a probabilistic framework with neighbor regularization and postprocessing mechanism to mine risk patterns from risk trajectories and meanwhile enable these three features.

Table 1. Math Notations for the Risk Trajectory Generation Model

Variable	Definition
M, m	m denotes the indicator of trajectories, $m \in \{1, \dots, M\}$.
N_m, n_m	n_m denotes the indicator of the points in the risk trajectory t_m, $n_m \in \{1, \dots, N_m\}$.
K, k	k denotes the indicator of risk patterns, $k \in \{1, \dots, K\}$.
V, v	v denotes the indicator of ST-Zones in a ST-Cube, $v \in \{1, \dots, V\}$.
z_{mn}	The zone indicator for the nth point in the risk trajectory t_m.
c_{mn}	The crowd score of the nth point in the risk trajectory t_m.
r_{mn}	The indicator of the pattern that generates the nth point of the risk trajectory t_m.
P_r	The selection probability of risk patterns for points in a given risk trajectory.
P_z	The selection probability of ST-zones for a given risk pattern.
P_c	The generation probability of crowd scores for a given risk pattern.
$\boldsymbol{\vartheta}^{(m)}$	The parameter of the multinomial distribution for pattern selection, i.e., the mixture proportion of risk patterns for t_m.
$\boldsymbol{\varphi}^{(k)}$	The parameter of the multinomial distribution for zone selection, i.e., the mixture proportion of ST-Zones in the risk pattern k.
$\boldsymbol{\psi}^{(k)}$	The parameters of the Beta distribution for the crowd score of ST-Zones in the risk pattern k.
ξ_v	The neighbor-based probability estimation of the ST-Zone s_v belonging to the K risk patterns.
$\boldsymbol{\alpha}$	The parameter of the prior Dirichlet distribution of $\boldsymbol{\vartheta}^{(m)}$.
$\boldsymbol{\beta}$	The parameter of the prior Dirichlet distribution of $\boldsymbol{\varphi}^{(k)}$.

4.1 Risk Trajectory Generation Model

In this section, we design a generative Bayesian network model, named ***Risk Trajectory Generation*** **(RTG)** model, to describe the generative process of risk trajectories. The RTG model focuses on describing the *frequency* feature of risk patterns. Here, a risk trajectory is assumed to be generated from a set of spatio-temporal patterns that were frequently visited by DGTs, and we estimate the parameters of the risk trajectory generation process to mine the hidden patterns from the risk trajectories.

4.1.1 Trajectory Generation Process. We here take a topic model view of the risk trajectory generation process by having a trajectory as a document and the ST-Zones in the trajectory as words, and having risk patterns as latent topics that generate the ST-Zones. Specifically, assuming there exist K risk patterns in the ST-Cube of a city, given a risk trajectory t_m with N points, the nth point of t_m, i.e., (z_{mn}, c_{mn}), is generated by the following three steps:

- *Step 1–Select* r_{mn}: We denote the risk pattern that generates the point (z_{mn}, c_{mn}) is $r_{mn} \in \{1, \dots, K\}$, which is one of the K risk patterns. To generate the point (z_{mn}, c_{mn}), the RTG model first selects r_{mn} from the K risk patterns. The probability of selecting the kth pattern as r_{mn} follows a trajectory-conditioned discrete distribution of K outputs, i.e., $P(r_{mn} = k|m)$. We further assume the distribution of $P(r_{mn} = k|m)$ is identical over different points in t_m. Hence, the risk patterns selected for the whole trajectory t_m are $\mathbf{r}_m = \{r_{m1}, \dots, r_{mN}\}$, which follows a trajectory-conditioned multinomial distribution (i.e., P_r) with $\{P(k|m)\}_{k=1}^{K}$ as the simplex parameters.

- *Step 2–Select* z_{mn}: For the nth point with $r_{mn} = k$, the model selects a ST-Zone from the ST-Cube as z_{mn}. The probability of selecting the vth ST-Zone (i.e., s_v) as z_{mn} follows a pattern-conditioned discrete distribution $P(z_{mn} = s_v|k)$. Or equivalently, we can characterize the zone selection by a multinomial distribution (i.e., P_z) with $\{P(s_v|k)\}_{v=1}^{V}$ as the simplex parameters.
- *Step 3–Generate* c_{mn}: For the nth point with $r_{mn} = k$, the crowd score $c_{mn} \in (0, 1)$ is generated from a pattern-conditioned continuous distribution $P_c(c_{mn}|k)$.

The above generation process indeed coincides with the real-world scenario of dangerous goods transportation. For a one-time transportation task, a transporter first plans a general route according to his/her experience, i.e., some risk patterns in mind, which corresponds to Step 1. During the transportation, the transporter visits specific ST-Zones covered by different risk patterns, which corresponds to Step 2. The crowdedness of the visited ST-Zones is then measured by the population density levels of the risk patterns that cover the ST-Zones, which corresponds to Step 3. The purpose of using a pattern-conditioned distribution for crowd score is to infer risk patterns in different crowdedness levels.

4.1.2 The Generative Bayesian Network Model. In this section, we choose specific types of distributions for the above-mentioned probabilities and build the Bayesian network model for risk pattern learning.

The trajectory generation process first selects r_{mn} using the probability P_r, and then selects z_{mn} using the probability P_z and generates c_{mn} using the probability P_c to form the point (z_{mn}, c_{mn}) of a risk trajectory. The probabilities P_r, P_z, P_c proposed in the trajectory generation process are described as follows:

$$\begin{aligned} P_r\left(k\middle|\boldsymbol{\vartheta}^{(m)}\right) &\sim Multinomial\left(\boldsymbol{\vartheta}^{(m)}\right), \\ P_z\left(s_v\middle|\boldsymbol{\varphi}^{(k)}\right) &\sim Multinomial\left(\boldsymbol{\varphi}^{(k)}\right), \\ P_c\left(c\,\middle|\boldsymbol{\psi}^{(k)}\right) &\sim Beta\left(\boldsymbol{\psi}^{(k)}\right). \end{aligned} \tag{3}$$

Here, the parameter $\boldsymbol{\vartheta}^{(m)} = \{\vartheta_1^{(m)}, \ldots, \vartheta_k^{(m)}, \ldots, \vartheta_K^{(m)}\}$ denotes the mixture proportion of the K risk patterns in the trajectory m. The parameter $\boldsymbol{\varphi}^{(k)} = \{\varphi_1^{(k)}, \ldots, \varphi_v^{(k)}, \ldots, \varphi_V^{(k)}\}$ denotes the mixture proportion of the V ST-Zones for the pattern k. $\boldsymbol{\psi}^{(k)} = \{\psi_\alpha^{(k)}, \psi_\beta^{(k)}\}$ is the parameter of the Beta distribution for the pattern k.

For different risk trajectories, the pattern mixture proportions should be different. We therefore introduce a Dirichlet prior to $\boldsymbol{\vartheta}^{(m)}$ as

$$P_\vartheta\left(\boldsymbol{\vartheta}^{(m)}|\boldsymbol{\alpha}\right) \sim Dirichlet\left(\boldsymbol{\alpha}\right), \tag{4}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_k, \ldots, \alpha_K\}$ is the hyper-parameter.

For different patterns, the ST-Zone mixture proportions should be also different, so we again introduce a Dirichlet prior to $\boldsymbol{\varphi}^{(k)}$ as

$$P_\varphi\left(\boldsymbol{\varphi}^{(k)}|\boldsymbol{\beta}\right) \sim Dirichlet\left(\boldsymbol{\beta}\right), \tag{5}$$

where $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_v, \ldots, \beta_V\}$ is the hyper-parameter.

According to the distributions in Equation (3) to Equation (5), the generative process of a risk trajectory can be modeled by a Bayesian network in Figure 3 (we here ignore the left-bottom regularization part of the plate). Given a risk trajectory $t_m = \{(z_{m1}, c_{m1}), \ldots, (z_{mN}, c_{mN})\}$, the joint

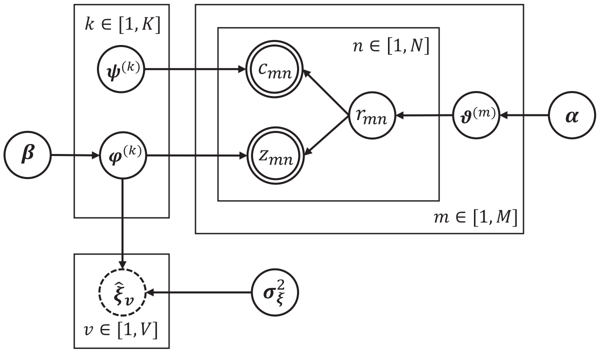


Fig. 3. The Bayesian network with the neighbor regularization for risk trajectory generation.

probability of the observed and hidden variables is

$$\begin{aligned} & P\left(z_m, c_m, r_m, \vartheta^{(m)}, \Phi, \Psi \middle| \alpha, \beta\right) \\ =& P_\vartheta\left(\vartheta^{(m)} \middle| \alpha\right) \cdot \prod_{k=1}^{K} P_\varphi\left(\varphi^{(k)} \middle| \beta\right) \cdot \prod_{n=1}^{N} \left[P_z\left(z_{mn} \middle| \varphi^{(r_{mn})}\right) P_c\left(c_{mn} \middle| \psi^{(r_{mn})}\right) P_r\left(r_{mn} \middle| \vartheta^{(m)}\right) \right], \end{aligned} \tag{6}$$

where $\Phi = \{\varphi^{(k)}\}_{k=1}^{K}$, $\Psi = \{\psi^{(k)}\}_{k=1}^{K}$, and z_m and c_m are vectors consisting of z_{mn} and c_{mn}, respectively.

Then, the likelihood probability of the trajectory point (z_{mn}, c_{mn}) with particular observation is

$$P\left(z_{mn}, c_{mn} \middle| \vartheta^{(m)}, \Phi, \Psi\right) = \sum_{k=1}^{K} P_z\left(z_{mn} \middle| \varphi^{(k)}\right) P_c\left(c_{mn} \middle| \psi^{(k)}\right) P_r\left(k \middle| \vartheta^{(m)}\right). \tag{7}$$

The likelihood of all trajectories $Z = \{z_m\}_{m=1}^{M}$ and corresponding crowd scores $C = \{c_m\}_{m=1}^{M}$ is

$$P\left(Z, C | \Theta, \Phi, \Psi\right) = \prod_{m=1}^{M} \prod_{n_m=1}^{N_m} P\left(z_{mn}, c_{mn} \middle| \vartheta^{(m)}, \Phi, \Psi\right), \tag{8}$$

where $\Theta = \{\vartheta^{(m)}\}_{m=1}^{M}$ is the set of parameters of all trajectories.

In Equation (8), the variables Z, C are observable, and Θ, Φ, Ψ are unobservable. We estimate the unobservable variables using the Gibbs sampling method [11], which is detailed in the Appendix. Here, risk patterns are characterized as a probabilistic form by the unobservable variables $\varphi^{(k)} \in \Phi$, which expresses a risk pattern as a distribution over all the ST-Zones in a ST-Cube.

The risk patterns expressed by $\varphi^{(k)}$ meet the frequency requirement of Definition 6 implicitly, because they are directly mined from the trajectory data of DGTs. However, the patterns might not meet connectivity and crowdedness requirements. We amend this by adopting neighbor regularization in Section 4.2 and pattern postprocessing in Section 4.3.

4.2 Spatio-temporal Neighbor Regularization

We here introduce a spatio-temporal neighbor regularization to our RTG model to meet the connectivity requirement.

Specifically, for each ST-Zone, we use the kernel density method to generate an *estimated probability* of the ST-Zone s_v belonging to a pattern k as

$$\xi_v^{(k)} = \text{softmax}\left(\sum_u \varphi_u^{(k)} g(s_v, s_u)\right), \tag{9}$$

where $g(s_v, s_u)$ is a Gaussian kernel density of the ST-Zone s_u to s_v. The expression of $g(s_v, s_u)$ is

$$g(s_v, s_u) = \exp\left(-\frac{\|\boldsymbol{v} - \boldsymbol{u}\|^2}{2\sigma^2}\right), \tag{10}$$

where the vectors $\boldsymbol{v} = (i_v, j_v, h_v)$ and $\boldsymbol{u} = (i_u, j_u, h_u)$ are the location indices of s_v and s_u in the ST-Cube, respectively. The parameter σ is the standard deviation of Gaussian kernels and is set as $\sigma = 1$ in practice.

We also define the *observed probability* of s_v belonging to the pattern k as

$$\hat{\xi}_v^{(k)} = \frac{d_{vk}}{\sum_{l=1}^K d_{vl}}, \tag{11}$$

where d_{vk} is the number of all risk trajectory points with $z_{mn} = s_v$ and $r_{mn} = k$. The variable z_{mn} can be obtained from the raw trajectory data, and r_{mn} can be obtained in Step 1 of the trajectory generation process.

We consider the variable $\hat{\xi}_v^{(k)}$ as a noisy observation of $\xi_v^{(k)}$ drawn from a Gaussian distribution with zero mean and variance σ_ξ^2 as follows:

$$P_\xi\left(\hat{\xi}_v^{(k)} \middle| \xi_v^{(k)}, \sigma_\xi^2\right) = \frac{1}{\sigma_\xi \sqrt{2\pi}} \exp\left(-\frac{\left(\hat{\xi}_v^{(k)} - \xi_v^{(k)}\right)^2}{2\sigma_\xi^2}\right). \tag{12}$$

Plugging the regularization into the risk trajectory generation model, the joint probability of all observed and hidden variables with given hyper-parameters is

$$\begin{aligned} & P\left(\boldsymbol{z}_m, \boldsymbol{c}_m, \boldsymbol{r}_m, \boldsymbol{\vartheta}^{(m)}, \Phi, \Psi, \Xi \middle| \boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma_\xi^2\right) \\ & = P_\vartheta\left(\boldsymbol{\vartheta}^{(m)} \middle| \boldsymbol{\alpha}\right) \cdot \prod_{k=1}^K P_\varphi\left(\boldsymbol{\varphi}^{(k)} \middle| \boldsymbol{\beta}\right) \cdot \prod_{n=1}^N \left[P_z\left(z_{mn} \middle| \boldsymbol{\varphi}^{(r_{mn})}\right) \right. \\ & \left. P_c\left(c_{mn} \middle| \boldsymbol{\psi}^{(r_{mn})}\right) P_r\left(r_{mn} \middle| \boldsymbol{\vartheta}^{(m)}\right) P_\xi\left(\hat{\xi}_{z_{mn}}^{(r_{mn})} \middle| \xi_{z_{mn}}^{(r_{mn})}, \sigma_\xi^2\right) \right], \end{aligned} \tag{13}$$

where $\Xi = \{\hat{\boldsymbol{\xi}}_v\}_{v=1}^V$. The probability graph of the regularized risk trajectory generative model is shown in Figure 3. The dash circle in the figure denotes the variables that are generated from parent variables using a deterministic function rather than a probability distribution. The hidden variables $\boldsymbol{\vartheta}^{(m)}$, Φ, Ψ and Ξ are inferred by Gibbs sampling, which is detailed in the Appendix.

We name this model the ***neighbor-regularized Risk Trajectory Generation*** **(nRTG)**, since it plugs a neighbor regularization into the RTG model. The neighbor regularization lets the ST-Zones with closer distances in ST-Cube have higher probability belong to the same risk pattern, so the nRTG model meets the connectivity requirement of Definition 6.

4.3 Risk Pattern Post-processing

In this section, the DGeye system postprocesses the probabilistic risk patterns expressed by $\boldsymbol{\varphi}^{(k)}$ by the following three steps:

- *Step 1–Pattern discretization*: This step binarizes $\boldsymbol{\varphi}^{(k)}$ to concentrate on several representative ST-zones. Specifically, given a risk pattern k, we rank ST-Zones by $P_z(s_v|k)$. Only the ST-Zones with a rank higher than a given threshold will be kept for the pattern. We call the binarized $\boldsymbol{\varphi}^{(k)}$ as raw patterns.
- *Step 2–Pattern splitting*: Although the nRTG model adopts a neighbor-regularization mechanism, it inevitably generates raw risk patterns containing some nonadjacent parts. Therefore, this step adopts a post-filtering scheme to split nonadjacent parts of a raw pattern into different patterns. This step could also generate some repeated patterns. We merge the repeated split patterns as a same pattern.
- *Step 3–Risk evaluation*: This step calculates the risk severity of each pattern. Intuitively, the risky level of a pattern is more serious if the general population density of its ST-Zones is higher. Therefore, for a risk pattern k, we use the expectation of its crowd score, i.e., $\mathbb{E}[P_c(c|\boldsymbol{\psi}^{(k)})]$, to indicate the risk severity. We call this expectation the *severity score* of the risk pattern. Patterns that have severity scores over some given threshold turn out to be the targeted risk patterns defined in Definition 6.

After postprocessing, a risk pattern mined by the DGeye system is a group of spatio-temporally adjacent ST-Zones with a severity score indicating the risk level.

5 RISK SOURCE TRACKING

In this section, we probe the forming reason of risk patterns to locate risk sources for administration. Indeed, risk patterns in a same city usually have some causal relationships among one another due to the "invisible hand" of supply-and-demand. For example, dangerous goods transportation across a residential area might be caused by the liquefied gas demand of a restaurant located in downtown. If we do not locate the demand source, i.e., the restaurant, but only forbid transporters to drive through the residential area, we just transfer the risk from one area to another.

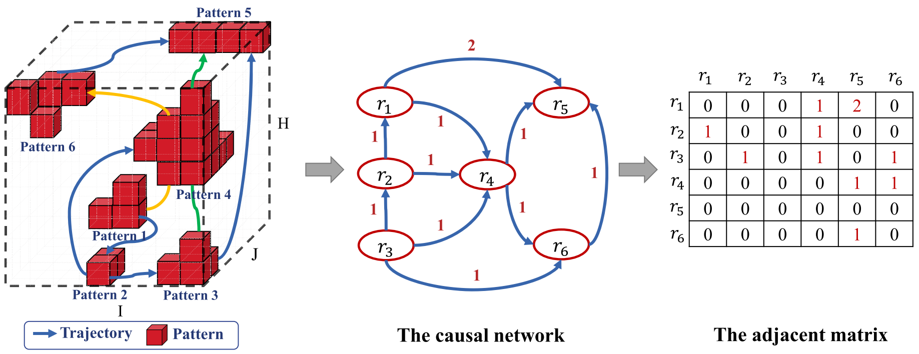
In what follows, we propose a ***Risk Source Tracking*** **(RST)** method to track the demand sources of dangerous goods forming the risk patterns. In short, RST first builds a causal network among risk patterns, and then applies random walk to rank the patterns. The top-ranked patterns are finally considered as the sources of DGT risks.

5.1 Building Risk Causal Network

RST first builds a weighted directed network, called the *DGT network*, to describe the influences among risk patterns. In the network, the vertices are risk patterns, and the edge weights are traffic volumes of DGT among the patterns.

Assume there are K risk patterns $R = \{r_k\}_{k=1}^{K}$. We express the adjacent matrix of the DGT network as $\boldsymbol{W} \in \mathbb{R}^{K \times K}$, where the element w_{xy} denotes the weight of edge from r_x to r_y. Given a set of DGT trajectories, w_{xy} is measured by the number of DGT driving from r_x to r_y (here, a DGT driving through a pattern means it passes at least one ST-Zone in the pattern). For a DGT passing through a sequence of patterns, we count the DGT to w_{xy} of any pair of the patterns in the sequence. Figure 4 shows a toy example for network building illustration. Since a DGT orderly drives through the patterns r_1, r_4, and r_5, the weights of $r_1 \rightarrow r_4$, $r_1 \rightarrow r_5$ and $r_4 \rightarrow r_5$ all need be increased by one.

In a DGT network, we regard r_y as a risk source of r_x if there exist vehicles carrying dangerous goods from r_x to r_y, since the reason a DGT has to pass r_x is there are dangerous goods (transportation) requirements in r_y. From this perspective, a DGT network is also a causal influence network among risk patterns. Therefore, we also name the DGT network as *Risk Causal Network* in our system.



	r_1	r_2	r_3	r_4	r_5	r_6
r_1	0	0	0	1	2	0
r_2	1	0	0	1	0	0
r_3	0	1	0	1	0	1
r_4	0	0	0	0	1	1
r_5	0	0	0	0	0	0
r_6	0	0	0	0	1	0

Fig. 4. An illustration of building a risk causal network.

5.2 Ranking Risk Patterns

Next, we rank the risk patterns according to their importance in the risk causal network. **A Random Walk with Restart (RWR)** algorithm [29] is applied for this task, with the assumption that a pattern that induces more and important patterns should have a higher rank of importance.

For the K patterns in the causal network, we define a ranking score vector $\boldsymbol{s} = (s_1, \ldots, s_k, \ldots, s_K)^\top$, where s_k is the score of pattern k. For the edge $r_x \rightarrow r_y$ with a weight w_{xy}, we define the causal transition probability g_{xy} as

$$g_{xy} = \frac{w_{xy}}{\sum_{k=1}^{K} w_{xk}}. \tag{14}$$

The RWR algorithm iteratively updates the ranking score vector $\boldsymbol{s}$ using a transition matrix $\boldsymbol{G} \in \mathbb{R}^{K \times K}$ composed of g_{xy}. In the $(\tau + 1)$-th round, $\boldsymbol{s}$ is updated as

$$\boldsymbol{s}^{(\tau+1)} = \alpha \boldsymbol{G} \boldsymbol{s}^{(\tau)} + (1 - \alpha)\boldsymbol{q}, \tag{15}$$

where $\boldsymbol{q}$ is the normalized risk severity vector of the risk patterns, with the kth element given by

$$q_k = \frac{\mathrm{severity}(r_k)}{\sum_{x=1}^{K} \mathrm{severity}(r_x)}, \tag{16}$$

where severity(r_x) is r_x's severity measured by the expectation of the distribution $Beta(\boldsymbol{\psi}^{(x)})$.

Note that we also use $\boldsymbol{q}$ to initialize $\boldsymbol{s}$, i.e., let $\boldsymbol{s}^{(0)} = \boldsymbol{q}$. It is easy to show that the above iterations will converge to the following steady state when $\tau \rightarrow \infty$ [45],

$$\boldsymbol{s}^* = (1 - \alpha)\boldsymbol{q}(\boldsymbol{I} - \alpha \boldsymbol{G})^{-1}, \tag{17}$$

which is finally adopted to rank the risk patterns. The pattern with a greater ranking score has a higher priority to be a risk source. A list of ranked risk patterns is valuable to the city's administrative department of dangerous goods. Urban planners can also optimize the functional layout of a city according to the list to eliminate the risk sources.

6 PATTERN RISKY STATE PREDICTION

Risk patterns and risk sources can indicate the general DGT risks existing in a city, which, however, is not adequate for real-time risk monitoring. In real-world applications, urban emergency services need to monitor the real-time state of risk patterns and predict their risky states to allocate emergency resources in a proactive manner. Therefore, the DGeye system adopts a *Risky Score* to indicate the risky state of risk patterns and provides a Bayesian network model to predict pattern states.

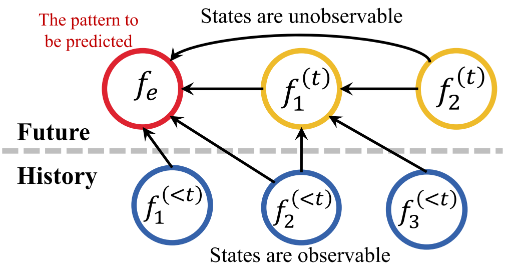


Fig. 5. An illustration of the predictive model.

6.1 Risky State of ST-Zones

In Section 3.2, we have defined the *Crowd Score* c_{ijh} to indicate the population density of a ST-zone s_{ijh}. We here introduce the concept of the *DGT Score*, denoted as d_{ijh}, to indicate the DGT density of s_{ijh}, i.e., the number of risk trajectory samples that cover s_{ijh}. Given c_{ijh} and d_{ijh}, we can now define the *Risky Score* of s_{ijh}, with the assumption that ST-zones with more crowded populations and transporters of dangerous goods tend to be more risky.

Let u_{ijh} denote the risky score of s_{ijh}, we have

$$u_{ijh} = D_M\left((d_{ijh}, 0)^\top, \mathbf{0}\right) \times D_M\left((0, c_{ijh})^\top, \mathbf{0}\right), \tag{18}$$

where $D_M(\cdot, \cdot)$ is the Mahalanobis distance. For two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ in a vector set, $D_M(\boldsymbol{a}, \boldsymbol{b})$ is defined as

$$D_M(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{(\boldsymbol{a} - \boldsymbol{b})^\top \Sigma^{-1} (\boldsymbol{a} - \boldsymbol{b})}, \tag{19}$$

where Σ is the covariance matrix of the vectors set. If u_{ijh} is greater than a threshold, then we say zone s_{ijh} is in a risky state (or a *Risky-Zone,* for short), otherwise in a safe state. In practice, the threshold is set to the 90% upper quantile of all the risky scores [30].

6.2 Pattern State Prediction Model

For a risk pattern, we say it is in the *risky state* at time t when it contains at least one risky-zone at time t, otherwise it is in the safe state. The task of pattern state prediction is to predict a risk pattern's state at t using the states of all patterns at time $< t$. For a risk pattern, there is only one risky state sample for each time slice. Therefore, the DGeye system adopts a Bayesian model for pattern state prediction, since the model is more suitable to the small sample scenario.

Given K risk patterns, we denote the state of pattern k at time t as $f_k^{(t)}$, where $f_k^{(t)} = 1$ for the risky state and 0 for the safe state. We denote the state of the pattern to be predicted at time t as f_e, group the states of other patterns at time t into an unobservable set $F = \{f_1^{(t)}, \ldots, f_k^{(t)}, \ldots, f_{K1}^{(t)}\}$, where $f_e \notin F$, and group the pattern states before t into an observable set $H = \{f_1^{(<t)}, \ldots, f_k^{(<t)}, \ldots, f_{K2}^{(<t)}\}$. It is reasonable to assume $f_k^{(t)}$ is only influenced by the states of patterns connected with r_k in the Risk Causal Network (see Section 5.1). The pattern state prediction problem can then be modeled by a Bayesian network, shown in Figure 5.

According to the Bayes' Theorem, the posterior probability of f_e conditioned on H and F is

$$\Pr(f_e \mid H, F) = \frac{\Pr(f_e) \Pr(H, F \mid f_e)}{\Pr(H, F)}, \tag{20}$$

which can be approximated by a naïve Bayesian method as

$$\begin{aligned}\Pr(f_e \mid H, F) &\propto \Pr(f_e) \prod_{k=1}^{K1} \Pr\left(f_k^{(t)} \mid f_e\right) \prod_{k=1}^{K2} \Pr\left(f_k^{(<t)} \mid f_e\right) \\ &\propto \ln\left(1 + \Pr(f_e)\right) + \sum_{k=1}^{K1} \ln\left(1 + \Pr\left(f_k^{(t)} \mid f_e\right)\right) + \sum_{k=1}^{K2} \ln\left(1 + \Pr\left(f_k^{(<t)} \mid f_e\right)\right),\end{aligned} \tag{21}$$

with Laplacian smoothing. Since the pattern states in F are unobservable, we cannot directly use Equation (21) to calculate the posterior probability of f_e. Therefore, we propose an **Expectation-Maximization (EM)** algorithm to estimate F and predict e in an iterative way. The influences among patterns in the risk causal network are also introduced into the algorithm.

The EM algorithm first initializes $f_x^{(t)}$ and f_e using an edge-weighted naïve Bayes model as follows:

$$\begin{aligned}f_x^{(t)}(0) &= \underset{f_x^{(t)} \in \{0,1\}}{\arg\max} \left[g_{xx} \ln\left(1 + \Pr\left(f_x^{(t)}\right)\right) + \sum_{k=1}^{K2} g_{kx} \ln\left(1 + \Pr\left(f_k^{(<t)} \mid f_x^{(t)}\right)\right) \right], \\ f_e(0) &= \underset{e \in \{0,1\}}{\arg\max} \left[g_{ee} \ln\left(1 + \Pr(f_e)\right) + \sum_{k=1}^{K2} g_{ke} \ln\left(1 + \Pr\left(f_k^{(<t)} k \mid f_e\right)\right) \right],\end{aligned} \tag{22}$$

where g_{kx} is the normalized causal influence between patterns r_k and r_x, as calculated in Equation (14).

In the τth round of the E-step, we update $f_x(\tau)$ using $f_e(\tau - 1)$, $F(\tau - 1)$ and H as

$$\begin{aligned}f_x^{(t)}(\tau) = \underset{f_x^{(t)} \in \{0,1\}}{\arg\max} &\left[g_{xx} \ln\left(1 + \Pr\left(f_x^{(t)}\right)\right) + g_{ex} \ln\left(1 + \Pr\left(f_e(\tau - 1) \mid f_x^{(t)}\right)\right) \right. \\ &\left. + \sum_{k=1}^{K2} g_{kx} \ln\left(1 + \Pr\left(f_k^{(<t)} \mid f_x^{(t)}\right)\right) + \sum_{k=1}^{K1} g_{kx} \ln\left(1 + \Pr\left(f_k^{(t)}(\tau - 1) \mid f_x^{(t)}\right)\right) \right].\end{aligned} \tag{23}$$

In the M-step, we predict f_e using H and $F(\tau)$ as

$$\begin{aligned}f_e(\tau) = \underset{f_e \in \{0,1\}}{\arg\max} &\left[g_{ee} \ln\left(1 + \Pr(f_e)\right) + \sum_{k=1}^{K2} g_{ke} \ln\left(1 + \Pr\left(f_k^{(<t)} \mid f_e\right)\right) \right. \\ &\left. + \sum_{k=1}^{K1} g_{ke} \ln\left(1 + \Pr\left(f_k^{(t)}(\tau) \mid f_e\right)\right) \right].\end{aligned} \tag{24}$$

When the algorithm reaches a stable state, we use the final $f_e(\tau)$ as the state prediction result. The prior probabilities and likelihoods are counted from the dataset.

7 EXPERIMENTS AND APPLICATIONS

In this section, we evaluate the effectiveness of the DGeye system over the DGT trajectory and mobile phone signaling datasets of two big cities in China: Beijing and Tianjin. In Section 7.1, we first introduce the datasets and the setups of experiments. In the next three subsections, we designed the experiments to evaluate the performance of the three components (i.e., Risk Pattern Mining, Risk Source Tracking, and Pattern State Prediction) of the DGeye system. In Section 7.2, we demonstrate the experiment results of the risk pattern mining component by visualizing the temporal feature and spatial distributions of the patterns. In Section 7.3, we evaluate the performance of the risk source tracking component through risk treatment simulation experiments and

Table 2. Statistics of Datasets after Pre-processing

Statistics	Beijing	Tianjin
#days	90	60
#vehicles	3,790	2,677
#GPS records	12,388,526	3,277,491
#trajectories	471,871	317,802
Average #grids an vehicle visits	522.6	407.2

case studies. Finally, in Section 7.4, we evaluate the effectiveness of the pattern state prediction component by comparing its prediction performance with several classic baselines.

7.1 Experimental Setup and Datasets

In the experiments, we apply the DGeye system for two big cities of China: Beijing[5] and Tianjin.[6] Beijing is the capital of China with a 21 million population, and Tianjin is a municipality directed by the central government with a 15 million population. The urban safety of the two cities is of the utmost importance undoubtedly. The datasets used in the experiments were collected from January 1 to March 31 in 2015 for Beijing, and from January 1 to March 1 in 2015 for Tianjin, including MPS records of population data and GPS records of road-specific vehicles transporting hazardous chemicals, fireworks, and civilian explosives. In the experiments, the system divides one day into 24 time slices, i.e., one hour per slice, and divides the maps of the two cities into $500m \times 500m$ urban zones, both resulting in 80×160 zones. Table 2 presents some detailed statistics of the datasets used in the experiments.

The types of dangerous goods in the dataset include hazardous chemicals, fireworks and crackers, and civil explosive materials. Hazardous chemicals refer to highly toxic chemicals and other chemicals that are toxic, corrosive, explosive, combustion-supporting, and so on, which are harmful to human installations and the environment. Typical civil explosive materials include gasoline and liquefied natural gas. The datasets do not contain the category information of the transport vehicles. According to the law of China, only vehicles that have special licenses are allowed to transport dangerous goods. In the experiments, we do not distinguish the types of dangerous goods and transport vehicles.

Figure 6(a) and Figure 6(b) show the spatial distributions of populations and DGTs in Beijing at 10:00 in one day. The colors indicate the distributional intensities: the redder, the higher. As can be seen, the populations of Beijing concentrate in the downtown area, but high DGT zones mainly locate at an outer beltway surrounding Beijing, i.e., the 5th ring road.[7] Figure 6(c) and Figure 6(d) exhibit the case of Tianjin at the same time slice. Obviously, the populations of Tianjin concentrate in two areas: the main urban area and the port area. The DGTs, however, are mainly distributed on the beltways and expressways connecting the port area and the main urban area.

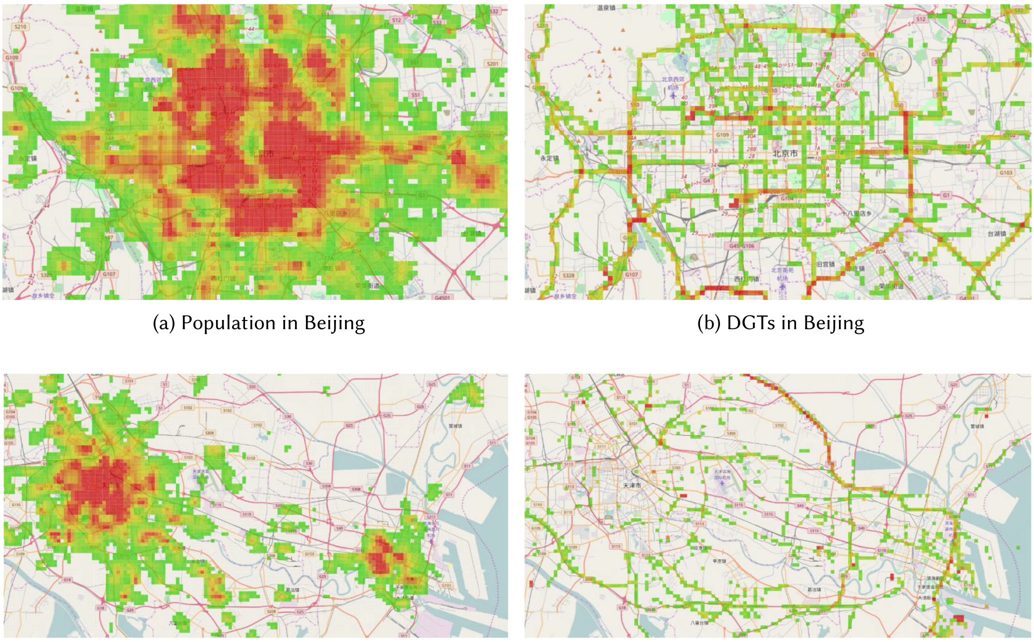
While Figure 6 indicates the distributional strengths of populations and DGTs do not resonate for both Beijing and Tianjin, the cities still expose to severe DGT risks given the painful historical lessons such as the massive explosion in Tianjin Port in August 2015.[8] That is why we consider to link the two variables together for risk pattern mining and risky state prediction.

[5] https://en.wikipedia.org/wiki/Beijing.

[6] https://en.wikipedia.org/wiki/Tianjin.

[7] https://en.wikipedia.org/wiki/5th Ring Road (Beijing).

[8] http://news.cntv.cn/special/video2015/tianjinbaozha/.



(a) Population in Beijing

(b) DGTs in Beijing

(c) Population in Tianjin

(d) DGTs in Tianjin

Fig. 6. Spatial distributions of residential populations and DGTs in Beijing and Tianjin.

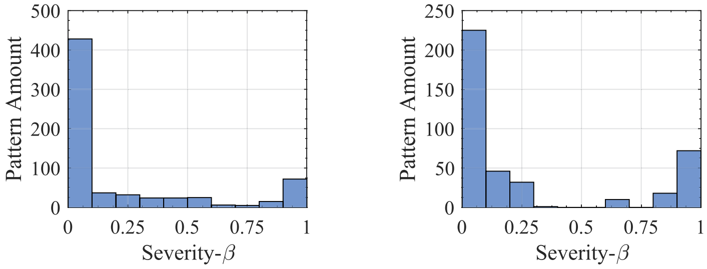
Table 3. The Number of Patterns That Are Split from Raw Risk Patterns

	nRTG		**RTG**	
City	Beijing	Tianjin	Beijing	Tianjin
Pattern #	671	407	1,320	517

7.2 Experimental Results of Risk Pattern Mining

We here demonstrate the risk pattern mining function of our system. In the experiments, we set the number of raw risk patterns as $K = 200$. K is a key parameter in the nRTG, which directly determines the number of risk patterns. Therefore, if we set K as a small number, then the nRTG model could not discover enough pattern. On the other side, we also find a large K will result in repeating risk patterns. Fortunately, the pattern repeat problem could be handled by the post-processing (see Section 4.3). In the second step of the post-processing, the nRTG model adopts a post-filtering scheme to split nonadjacent parts of a raw pattern into different patterns. If we set K as a large size, then the model will generate many repeat patterns in this post-processing step. To handle this problem, we merge the repeated split patterns as the same pattern. Taking advantage of this mechanism, the nRTG could handle the pattern repeat problem caused by setting K as a too-large size. In our experiments, we set K to 200, which is large enough in our datasets according to a trial and error method.

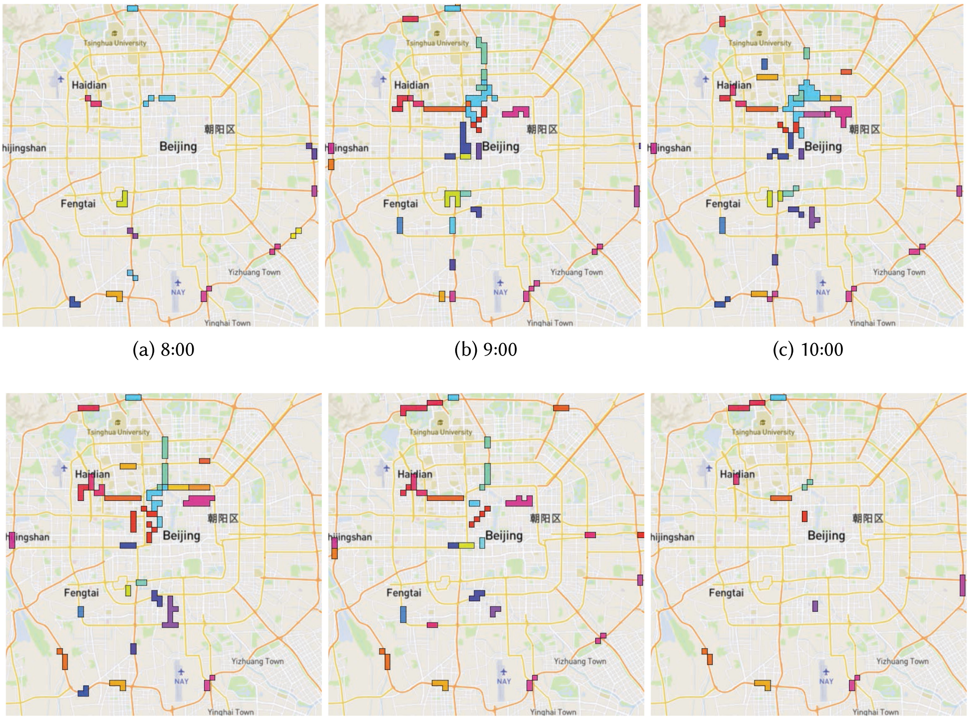
The number of output patterns is larger than this number due to the pattern splitting in post-processing (see Section 4.3). Table 3 gives the numbers of patterns mined from the Beijing and Tianjin datasets. As a bigger city, Beijing has more risk patterns than Tianjin.



(a) Beijing

(b) Tianjin

Fig. 7. The numbers of risk patterns with different β-severity.



(a) 8:00

(b) 9:00

(c) 10:00

(d) 11:00

(e) 12:00

(f) 13:00

Fig. 8. Spatial distribution of high-severity risk patterns in Beijing: (a) to (f) are the risk patterns' maps in Beijing from 8:00 to 13:00 AM.

In our system, each pattern has a Beta distribution to describe its severity score, denoted as β-severity for concision, stemming from populations. Figure 7 shows the β-severity distributions of the risk patterns for Beijing and Tianjin, respectively. As can be seen, the distribution of β-severity has a long tail for both cities, and the high β-severity patterns in the tail indeed need special attentions. In Figure 8 and Figure 10, we plot the risk patterns whose β-severity values are higher than the average. Figure 8 is for Beijing and Figure 10 for Tianjin. Since the risk patterns are 3D manifolds in the ST-Cube, we visualize them as 2D shapes on urban maps along the horizontal axis of time slice, which exhibits the dynamic evolution of the risk patterns over time.

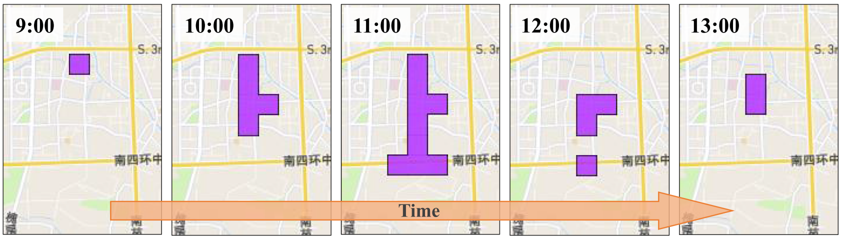


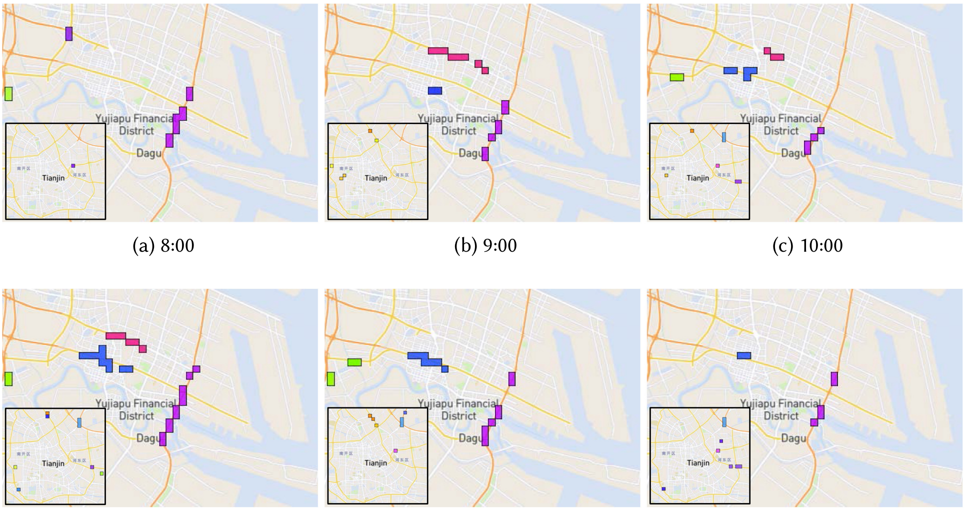
Fig. 9. An illustration of the evolution of risk pattern in Beijing.

Let us take a close look into Figure 8 for the risk patterns in Beijing with β-severity > 0.2 from 8:00 to 13:00. We use colors to distinguish different patterns. As can be seen, most of the high severity patterns located in the downtown area of Beijing, which is very different from the DGT distribution over the 5th ring road in the suburb areas in Figure 6(b). The reason for this difference is that DGEYE considers both DGT and population at the same time for risk pattern mining. That means while there exist many DGTs driving over the suburb roads, only those passing through densely populated areas would lead to risks. Some unknown risks in Figure 6 (especially in the downtown areas) are discovered by our model in Figure 8. This indeed validates the effectiveness of our probabilistic mining model.

Furthermore, Figure 8 demonstrates the dynamics of risk patterns. As can be seen, there exists no high-severity pattern in the downtown area at 8:00. The patterns are gradually increasing in the downtown area from 8:00 to 10:00, and gradually decreasing from 11:00 to 13:00, which indicates an obvious "tide" in the pattern evolution process. We can also perceive the "tide" by watching how a typical risk pattern goes through the germination, prosperity, and decay periods. For instance, the risk pattern in Figure 9 appears at 9:00 with a small size, then becomes a big one from 9:00 to 11:00 and gradually disappears from 11:00 to 13:00. Compared to our previous work in Reference [30], where the risk patterns only contain one time slice, our new approach can reveal more dynamic information of urban risks.

Figure 10 plots the dynamics of risk patterns from 8:00 to 13:00 in the port area of Tianjin, and the left-bottom small maps show the patterns of the downtown area. Different from Beijing, most of big-size patterns in Tianjin are in the port area rather than downtown area. The reason lies in the distinct requirements of dangerous goods of the two cities. The demand for dangerous goods in Beijing is people's daily consumption, such as gasoline for gas stations and liquefied gas for restaurants densely located in the downtown area. In consequence, DGTs have to drive through many densely populated zones in downtown, which results in big-size risk patterns appearing in the center area of Beijing. In contrast, the dangerous goods for Tianjin are mostly chemical materials for import and export trade. Therefore, most of the big-size patterns are generated in the port area of Tianjin.

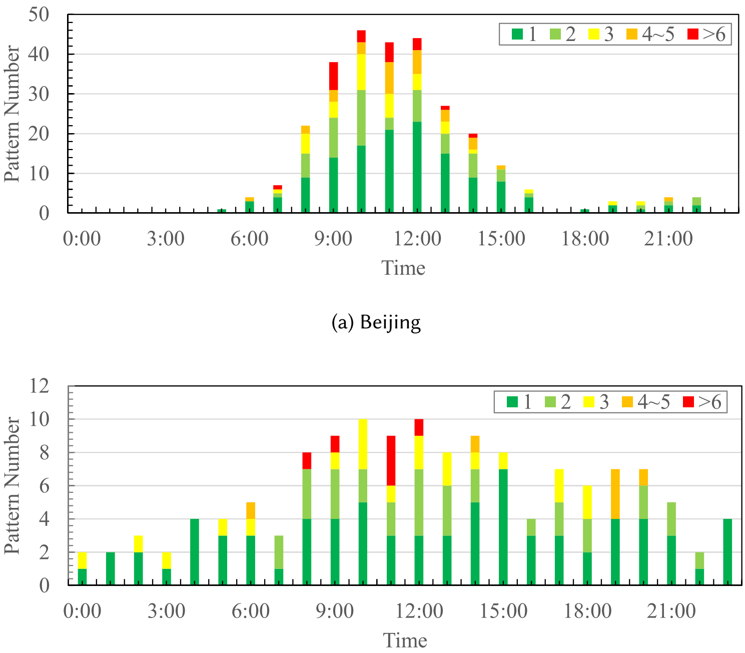
The risk patterns of Tianjin also have different temporal distribution compared with that of Beijing. Figure 11 shows the numbers of risk patterns for Beijing and Tianjin, respectively, along time slices, with different colors indicating diverse pattern sizes (i.e., the number of ST-Zones in a pattern). As can be seen, the temporal distribution of risk patterns has an obvious tide in Beijing but is much more smooth in Tianjin. Moreover, Beijing has more big-size patterns compared with Tianjin. The two differences again verify the distinct types of dangerous goods requirements of the two cities and sheds light on making differentiated DGT monitoring policies. That is, the Beijing government should pay more attention to DGTs in the downtown area in the middle of a day, while the Tianjin government should keep close watch over the port area for a longer time period.



(a) 8:00 (b) 9:00 (c) 10:00

(d) 11:00 (e) 12:00 (f) 13:00

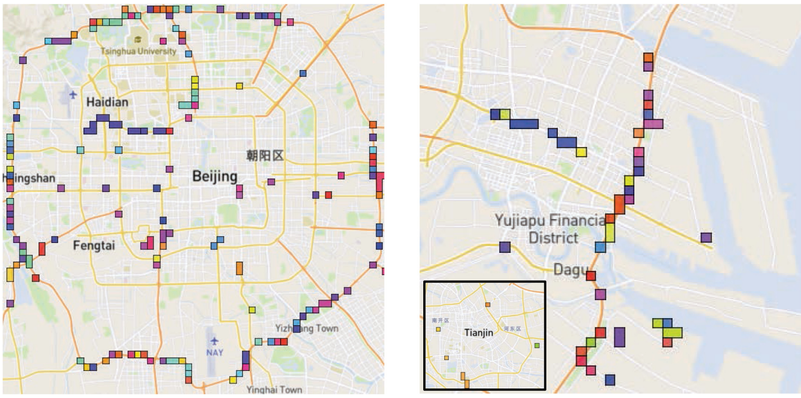
Fig. 10. Spatial distributions of high-severity risk patterns in Tianjin: (b) to (e) are the risk patterns' maps in Tianjin from 09:00 to 12:00 AM.



(a) Beijing

(b) Tianjin

Fig. 11. Temporal distributions of risk patterns in Beijing and Tianjin.



(a) Beijing, 9:00 (b) Tianjin, 9:00

Fig. 12. Mining risk patterns using the RTG model (without the neighbor regularization).

We finally compare the risk patterns generated with and without neighbor regularization. Figure 12 plots the risk patterns mined by the RTG model, i.e., our nRTG model without the neighbor regularization, for both Beijing and Tianjin at 9:00. By comparing Figure 12(a) with Figure 8(b) and Figure 12(b) with Figure 10(b), respectively, we can observe that the patterns mined by RTG are much more fragmented than those by nRTG. This also coincides with the fact that in Table 3 RTG generates much more risk patterns than nRTG. This indicates that the connectivity of the raw patterns generated by nRTG is better than that by RTG, which can help to reduce administrative workloads for false alarms. In short, this experiment verifies the effectiveness of the neighbor regularization mechanism in our nRTG model.

7.3 Experimental Results of Risk Source Tracking

7.3.1 Risk Treatment Simulation. The function of the RST algorithm in Section 5 is to rank the importance of risk patterns and urban governments can use the ranking as a priority list for DGT risk treatment. In real-world DGT risk management, the patterns with high rankings should be treated preferentially. Since the high-priority patterns in the RST ranking are risk sources of other patterns, if the high-priority patterns were treated, the risks in other patterns should also reduce. Therefore, it could be considered as quantitative performance merit of the RST algorithm that the average risk-reducing of all ST-Zones after top-L patterns were treated.

In this section, we give a simulation experiment to demonstrate the effectiveness of our pattern ranking algorithm using this performance merit. The experiment runs a simulated risk treatment over the Beijing and Tianjin datasets. Specifically, we apply the ranking algorithm on the data of first 2/3 days, i.e., 60 days for Beijing and 40 days for Tianjin, to get pattern rankings, and then artificially remove the DGT trajectories whose destinations are within the top-L ranking patterns. After removing the trajectories, we calculate the average *risky score* (see Equation (18)) of the ST-Zones in a risk pattern or in a particular area of a city on the rest days of the data. The reduced percentage of the new risky score compared with that before the trajectory removal is considered as a performance estimation of the RST algorithm. We also conduct the paired t-test [12] between RST and other baselines on the simulation results of each day to test the significance of improvement.

Table 4. Results of Risk Pattern Treatment Simulation for Beijing and Tianjin

	Panel A: Reduce of Average Risky Scores in Beijing						
Patterns	Area	Method					
		RST	PRW	Risky-Score	DGT-Score	β-severity	Crowd-Score
Top 5	Patterns	**45.52%****	41.40%	21.00%	15.05%	14.17%	6.09%
	Downtown	**24.51%****	20.33%	10.42%	10.59%	6.51%	3.42%
	Entire City	**20.25%****	13.08%	8.40%	9.88%	5.19%	2.72%
Top 10	Patterns	**67.20%****	62.10%	35.09%	40.90%	29.55%	14.09%
	Downtown	**37.30%****	32.14%	18.08%	22.72%	14.01%	7.57%
	Entire City	**31.60%****	26.30%	14.81%	19.75%	11.11%	6.42%
Top 20	Patterns	**73.88%****	65.02%	51.05%	58.24%	35.44%	20.77%
	Downtown	**42.75%****	35.10%	28.01%	32.49%	18.81%	11.89%
	Entire City	**37.53%****	31.49%	23.21%	29.14%	14.81%	10.12%
Top 50	Patterns	**82.23%****	73.76%	70.93%	69.84%	45.67%	45.75%
	Downtown	**51.22%***	48.02%	41.69%	41.45%	26.71%	26.38%
	Entire City	**44.94%***	42.10%	35.06%	37.28%	21.98%	21.98%
Panel B: Reduce of Average Risky Scores in Tianjin							
Patterns	Area	Method					
		RST	PRW	Risky-Score	DGT-Score	β-severity	Crowd-Score
Top 5	Patterns	**59.14%****	46.15%	39.68%	5.30%	8.62%	15.06%
	Downtown	**26.30%****	24.13%	22.00%	0.80%	1.90%	7.40%
	Port Area	**38.68%****	32.05%	19.34%	3.54%	5.66%	11.32%
	Entire City	**27.71%****	22.11%	18.07%	2.01%	3.21%	6.83%
Top 10	Patterns	**68.80%****	63.00%	40.72%	24.34%	12.64%	26.35%
	Downtown	**28.70%****	23.03%	22.20%	8.40%	2.80%	14.40%
	Port Area	**42.92%****	37.16%	19.58%	13.21%	8.49%	15.57%
	Entire City	**32.53%****	26.01%	18.88%	10.04%	4.82%	12.05%
Top 15	Patterns	**76.11%****	60.05%	43.11%	44.01%	23.96%	34.90%
	Downtown	**33.80%****	28.00%	25.90%	18.90%	10.80%	20.90%
	Port Area	**50.47%****	44.04%	20.52%	22.88%	12.97%	20.05%
	Entire City	**37.35%****	31.12%	20.48%	18.88%	10.84%	17.27%
Top 20	Patterns	**80.47%****	76.70%	69.77%	51.35%	48.51%	54.67%
	Downtown	**39.50%****	35.38%	31.70%	24.90%	27.50%	33.30%
	Port Area	**56.84%****	51.22%	41.04%	29.25%	28.77%	34.67%
	Entire City	**43.37%****	38.69%	31.73%	24.10%	24.50%	28.51%

Significantly outperforms PRW at the: ** 0.01 and * 0.05 level, paired t-test.

Table 4 shows the results. Here, we compare the RST algorithm with three baselines:

- *Risky-Score:* ranking risk patterns according to the average risky score of ST-Zones in a pattern.
- *DGT-Score:* ranking risk patterns according to the average DGT score of ST-Zones in a pattern.
- *β-severity Score:* ranking risk patterns according to the mean of Beta distribution of a pattern.
- *Crowd-Score:* ranking risk patterns according to the average crowd score of ST-Zones in a pattern.

The score definitions can be found in Section 3 and Section 6.1. Besides, we also use the ranking algorithm of our previous work [30] as a baseline, named PRW. Especially, we divide the patterns discovered by nRTG model as several sub-patterns by time slices. In this way, the sub-patterns are in the same format as the risk patterns discovered by our previous work [30]. We adopt importance ranking method in Reference [30] to calculate the ranking score of each sub-patterns, and we rank risk patterns using the summation of its sub-patterns.

In the simulation experiment, we remove trajectories with destinations in the top 5, 10, 20, 50 ranking patterns, respectively. From the table, we can observe that:

- The RST algorithm achieves the best performance compared with the best baseline under a significance level of 0.01 for most of the scenarios, which verifies the effectiveness of our approach. For both Beijing and Tianjin, using the RST-based treatment strategy, we can reduce the overall risk level of risk patterns by 50% even only the top-five patterns are treated. The reduced proportions for the entire city are more than 20% for both Beijing and Tianjin, which is a significant performance for effective urban risk treatment and implies much lower administrative costs as well.
- The second best algorithm is our previous work [30]. It is much better than other baselines. This indicates causal network and RWR framework adopted by both RST and our previous work is an effectiveness approach to discover important risk patterns. The performance of PRW is less than the RST. This verifies the effectiveness of the new pattern mining method proposed in this article. The patterns discovered by nRTG can cross multiple time slices, so it provides a more complete coverage of DGT risks in a city. Therefore, the nRTG-based risk source tracking algorithm has better performance.
- The third-best algorithm is Risky-Score, but its performance even cannot reach a half of that of the RST algorithm. The reason is that the Risky-Score approach only pays attentions to high-risk patterns but cannot maximize the treatment effect through exploiting the causal relationships among risk patterns. This in turn validates the importance of the risk causal network proposed in Section 5.1.
- DGT-Score performs better than β-severity Score and Crowd-Score. This indeed agrees with our intuition that it is the DGTs rather than populations that trigger the urban risks. The DGT trajectories rather than the residential areas are the key to probing the risks.
- The performance of β-severity is better than Crowd-score in Beijing and worse than Crowd-score in Tianjin. It is understandable that the performance of β-severity is comparable with Crowd-score, since both β-severity and Crowd-score stem from populations.

In summary, the experiment results demonstrate that our RST algorithm can indeed locate highly influential risk patterns in a city, which is of great use to setting treatment priority for risk pattern management given limited administrative resources.

7.3.2 *Cases of Risk Sources Tracked.*

Here, we verify the effectiveness of the RST algorithm by illustrating two real-world cases of risk sources traced by our approach.

Figure 13 shows the No. 1 risk pattern (the red block in the map) detected by RST in Beijing. As can be seen, this pattern is located at the Dongzhimen and Dongsi district, which is a famous entertainment district of Beijing.[9] Especially, the Dongzhimen area has an extremely well-known food street, Guijie.[10] A major cuisine ordered by crowded diners in Guijie is the "hotpot," which is a kind of cuisine that cooks raw foods in a simmering metal pot at the center of dining tables. A hotpot table is usually equipped with a mini gas stove connected to a liquid gas cylinder, which forms

[9] https://en.wikipedia.org/wiki/Dongzhimen.

[10] https://www.travelchinaguide.com/attraction/beijing/guijie-street.htm.

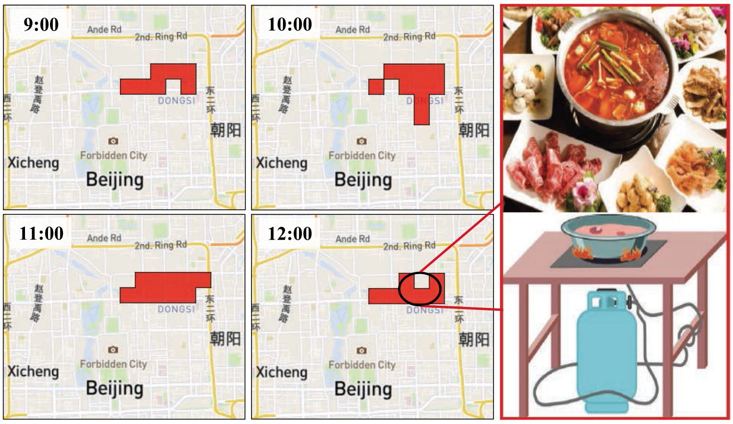


Fig. 13. The No. 1 risk pattern provided by RST in Beijing.

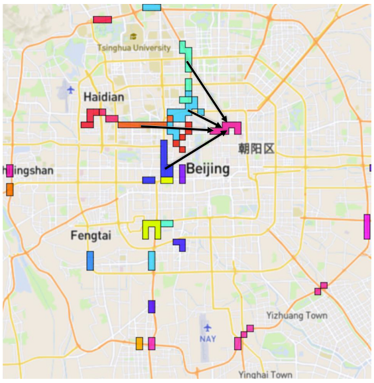


Fig. 14. The types and relations of the risk patterns in Beijing at 9:00.

an enormous demand of gas cylinders transported by DGTs to Guijie every day. This finding was reported to the Beijing government as a report of the DGeye system. The Beijing government later launched a gas pipeline laying project in Guijie in September 2016. As reported by the media [23], Beijing "Guijie" finally bid farewell to the gas-cylinder era in 2017.

Figure 14 gives the risk pattern distribution at 9:00 in Beijing. As shown in the figure, there is a pink color pattern on the upper right part of the map, which is the top-1 source pattern of DGT risks in Beijing. In Figure 14, we connect several patterns to the pink pattern using black arrows. These patterns are downstream patterns of the pink pattern in the pattern causal network. We can see these patterns cover the roads from suburb to the pink pattern. In addition to the above patterns, we also see there are some patterns located over the outer ring road of Beijing. These patterns are gathering points of dangerous goods transporters entering the city. From the figure, we can see at least three types of patterns, i.e., destination of DGTs, the area that must be passed by DGTs to their destination, and gathering points of entering the city. The three types of patterns construct a logistics chain of dangerous goods in Beijing city. If we want to treat the risk caused by the DGT logistics chain, removing the demands for dangerous goods in the risk source pattern is the most fundamental method. That is why the RST algorithm can achieve superior risk treatment performance compared with other priority ranking algorithms.

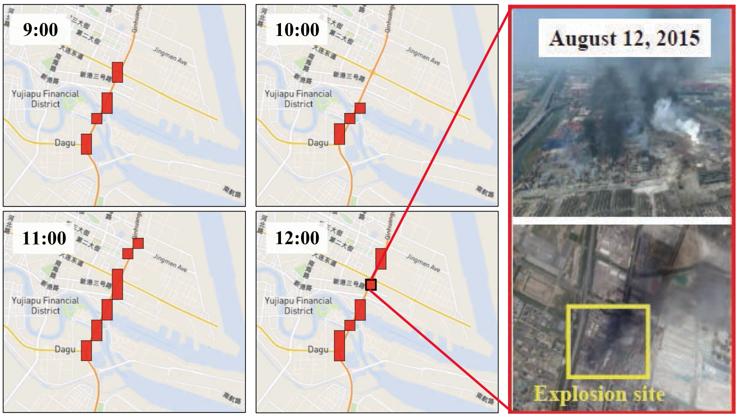


Fig. 15. The No. 1 risk pattern provided by RST in Tianjin.

Figure 15 shows the No. 1 risk pattern in Tianjin. We can see this pattern resides in the port area, along with a north-south oriented road aside a wharf. Around this pattern is a financial district crowded with residents, which reveals a hidden defect of urban planning in this area: the depots of dangerous goods in the wharf are too close to the residential areas. This fatal defect actually triggered an irreparable tragedy: the Tianjin port explosion of dangerous goods on August 12, 2015, which happened right at a road intersection covered by the No. 1 risk pattern, as shown in the bottom-right sub-figure of Figure 15.

7.4 Experimental Results of Pattern State Prediction

In this section, we evaluate the pattern state prediction performance of the DGeye system. The datasets of Beijing and Tianjin contain trajectories and mobile-phone records of 90 and 60 days, respectively. We use data in the first 2/3 days (60 days for Beijing and 40 days for Tianjin) as the training set and the rest as the test set. The following baseline methods are included for comparison:

- *Likelihood model (LL)*, which uses the likelihood probability to predict the state of a pattern at a given time slice. Given a pattern state f_e to be predicted, we calculate the $\Pr(f_k^{(<t)} \mid f_e)$ for each pattern from training dataset as

$$\Pr(f_k^{(<t)} \mid f_e) = \frac{\Pr(f_k^{(<t)}, f_e)}{\Pr(f_e)}. \tag{25}$$

Then, we use the likelihood $\sum_{k=1}^{K2} g_{ke} \ln(1 + \Pr(f_k^{(<t)} \mid f_e))$ in Equation (22) to predict pattern states.

- *Naive Bayes model (NB)*, which uses the initialization model of the EM algorithm in Equation (22) for prediction.
- *EM without casuality (EM¬g)*, which uses the EM model without the weight g_{ke} in Equation (22)–Equation (24) to predict the pattern state f_e. This baseline is used to evaluate the value of pattern causal relationships in the prediction.

It is easy to note that LL, NB, and EM¬g are the simplified versions of our proposed EM model. We use them as references to verify the effectiveness of the disabled components of the EM model.

- *Logistic Regression model (LR)*, which uses $g_{ke} f_k^{(<t)}$ as features of the pattern state f_e and trains a logistic regression to predict the pattern state f_e at time t.

Table 5. Prediction Performance Comparison for the Beijing Dataset

	EM	EM¬g	PRW	NB	LL	LR	SVM	ANN
Pr-risk	**0.742**	0.736	0.732	0.728	0.543	0.727	0.724	0.688
Re-risk	**0.768***	0.756	0.748	0.738	0.302	0.720	0.736	0.687
F1-risk	**0.754***	0.746	0.740	0.733	0.388	0.723	0.730	0.687
Pr-safe	**0.720***	0.708	0.701	0.692	0.467	0.680	0.688	0.706
Re-safe	0.691	0.682	0.671	0.707	0.687	0.687	0.675	**0.707**
F1-safe	0.705	0.687	0.686	0.698	0.562	0.683	0.682	**0.707**
Pr-all	**0.732***	0.723	0.719	0.711	0.508	0.705	0.707	0.697
Re-all	**0.732***	0.724	0.718	0.712	0.489	0.704	0.707	0.697
F1-all	**0.732***	0.724	0.719	0.712	0.498	0.705	0.707	0.697

Significantly outperforms EM¬g at the: * 0.05 level, paired t-test.

Table 6. Prediction Performance Comparison for the Tianjin Dataset

	EM	EM¬g	PRW	NB	LL	LR	SVM	ANN
Pr-risk	**0.590***	0.585	0.578	0.552	0.541	0.528	0.559	0.505
Re-risk	0.648	0.571	0.567	0.346	0.622	0.643	**0.657**	0.590
F1-risk	**0.618****	0.603	0.574	0.562	0.422	0.580	0.604	0.545
Pr-safe	**0.774***	0.763	0.755	0.736	0.676	0.751	0.767	0.724
Re-safe	0.728	0.721	0.718	**0.823**	0.734	0.653	0.686	0.650
F1-safe	**0.750**	0.748	0.736	0.728	0.742	0.698	0.724	0.685
Pr-all	**0.705***	0.696	0.687	0.667	0.625	0.667	0.689	0.641
Re-all	**0.698**	0.692	0.685	0.664	0.643	0.649	0.675	0.628
F1-all	**0.701**	0.694	0.686	0.665	0.634	0.658	0.682	0.634

Significantly outperforms EM¬g at the: ** 0.01 and * 0.05 level, paired t-test.

- *Support Vector Machine model (SVM)*, which uses $g_{ke}f_k^{(<t)}$ as features of the pattern state f_e and trains an SVM model to predict the pattern state f_e at time t.
- *Artificial Neural Networks model (ANN)*, which also uses $g_{ke}f_k^{(<t)}$ as features and trains a neural network model with two hidden layers and each layer has 30 neurons to predict the pattern state f_e at time t.
- *Previous Work (PRW)*, which divides the patterns discovered by nRTG model as several sub-patterns by time slices, and adopts the method in Reference [30] to predict pattern states.

Table 5 and Table 6 give the prediction performance of our EM algorithm and the baselines. Here, the evaluation metrics include precision (Pr), recall (Re), and F1 scores (F1) for the risky state, safe state, and both. We replicate 30 times for each experiment to conduct the paired t-test [12] and the mean values of each metrics are presented. As the tables show:

- For both Beijing and Tianjin datasets, the proposed model (*EM*) performs the best in most of the cases. Specifically, *EM* outperforms the second-best algorithm (the EM¬g baseline) under a significance level of 0.01 for most cases, which generally verifies the effectiveness of our algorithm.
- The performance of EM is better than the NB baseline. Since the EM model takes unobservable real-time pattern states $f_k^{(t)}$ into consideration but the NB model does not, this result verifies the information value of the unobservable states in prediction.

- The performance of EM is better than the EM¬g baseline. Since the latter is a simplified EM model that does not consider causality among patterns, this result again verifies the importance of pattern causal relationships for pattern state prediction.
- The performance of EM and EM¬g is better than the PRW baseline. This verified the new pattern mining method proposed by this article is more effective than the previous version of DGEYE.
- LR, SVM, and ANN perform relatively poor. This might be ascribed to the scarcity of training samples; that is, we can only sample the state of f_e one time in one day, and hence only have 60 and 40 training samples for every pattern state for Beijing and Tianjin, respectively. In this situation, Bayesian methods seem more effective than completely supervised classification models.

To sum up, our EM-enabled Bayesian model is very suitable for pattern risky state prediction by incorporating the real-time information of unobservable pattern states and the causal relation information among risk patterns.

8 RELATED WORK

Dangerous Goods Transportation. Dangerous goods is an important topic in hazardous materials management and **intelligent transportation systems (ITS)** [6]. To control societal risks caused by dangerous goods, some DGT monitoring systems, like MITRA [25], are deployed [5]. Most of them focus on monitoring and collecting locations of DGTs only but omit the important human activities. In academia, most ITS researchers focus on DGT route planning [24] and transportation systems design such as railway DGT [26]. In hazardous materials management, researchers focus on DGT risk assessment [14] and analysis [18]. Most of these works study DGT from an operations and optimization view and have a basic assumption: If a plan is well designed and deployed, DGT risks will be under control. In practice, however, many uncertainties could disturb the deployment of plans. Data-driven approaches are thus becoming more desired to detect and analyze risks of dangerous goods in real-world applications.

Spatio-temporal Pattern Mining. Mining latent patterns from spatio-temporal datasets is a popular area in the data mining community. Many spatial clustering algorithms, such as DBSCAN [7], and ST-DBSCAN [2], generate spatial patterns from a spatial distance view [9]. The collocation [15] and spatio-temporal sequential patterns mining [4, 16] algorithms detect frequent collocations and/or concurrences from spatio-temporal datasets. Our previous work [30] proposes an Apriori-like algorithm for mining spatial risk patterns from dangerous goods transportation and human activity data. The probabilistic graph-based topic model is another type of widely used methods for spatio-temporal pattern mining. One stream of such research is to discover urban activity rhythms from human mobility data, like geo-location-based check-in data or trajectory data. For instance, in Reference [10], the authors explore urban activity patterns from check-in data of social media based on topic models. In Reference [27], the authors use topic model to explore the evolution of urban traffic dynamic patterns from **Origin-Destination (OD)** data. Another research stream applies probabilistic graph-based methods that combine human mobility spatio-temporal data with some static domain knowledge for urban region function inference [40, 41], location recommendation [13, 22], regional health level prediction [38], urban culture pattern exploration [44], and still more. The risk pattern mining method in DGEYE is also based on probabilistic graph models, but the goal of mining spatio-temporal patterns from multivariate risk trajectory data in this article leads to a completely different model with crowd score generation and neighbor regularization.

Transportation Causality Analysis and Prediction. Transportation causality analysis and prediction are also the key functions of DGEYE. In the transportation causal analysis area, most of works focus on discovering causal relations in regular urban transportation. For instance, Reference [21] proposes an outlier tree–based causality discovery algorithm for spatio-temporal interactions in urban traffic data. In Reference [31], the authors adopt an interpretable deep learning method to discover inferences between road segments. Reference [3] proposes a two-step framework for inferring the root cause of anomalies in urban traffic data. However, few works have analyzed causality for cargo transportation. In the transportation prediction area, most of works focus on traffic speed prediction [19, 32] and traffic flow prediction [1]. Recent works in this area are devoted to designing complex deep learning network structures to achieve high-performance traffic speed/flow prediction [37], route recommendation [35] and trajectory recovery [34]. These solutions are not suitable to small example applications such as the risk pattern prediction in this article.

Urban Computing. Our work also falls into the research category of urban computing [43]. Besides the above-mentioned works on urban discovery, there are many others related to our study, including: data-driven urban analysis [28, 33], urban anomaly detection [42], urban public security [17], citizen behavior prediction [20], road network representation [39], and still more. To our best knowledge, our work is among the earliest studies in urban computing area that try to snuff out the threats from dangerous goods.

9 CONCLUSION

In this article, we present a novel system called DGeye for urban dangerous goods management. DGeye features in leveraging both DGT trajectory data and human activity data for risk tracking and monitoring. Specifically, DGeye discovers spatio-temporally adjacent risk patterns in a city by a carefully designed neighbor-regularized Risk Trajectory Generation model. Taking risk patterns as basic units, a Risk Causal Network is built for risk source tracking, and an EM-enabled algorithm is designed for risky state prediction, which makes DGeye an ideal decision support system for urban DGT risk management. In the introduction, we raised three research questions of this study: (1) how to identify risk patterns, (2) how to model relations among the patterns, and (3) how to predict the state of the patterns. In the DGeye system, the three questions have been answered by the three components of the system, i.e., the nRTG model for question (1), Risk Causal Network for question (2), and the EM-enabled prediction algorithm for question (3). DGeye has proven itself in the successful deployment for DGT risk management in two cities: Beijing and Tianjin. In particular, the report from DGeye has driven the Beijing government to lay down a gas pipeline in famous food street: Guijie, which finally bid farewell to the long history of gas-cylinder usage.

As for future work, we will consider extending our system from three directions. The first is to consider the types of dangerous goods. In this version of the DGeye, we consider all dangerous goods as the same type. In future work, we could introduce the type of dangerous goods and their level of danger in our nRTG model. The second direction is to extend the system as a human-in-the-loop mode, where the system could update the causal network and the risk ranks of patterns after a person removes a pattern in risk treatment simulation. In this way, the system could be more suitable for decision-making scenarios. The third direction is to introduce the dangerous goods transporter scheduling function in our DGeye system. This function is indeed needed in urban dangerous goods transportation management.

APPENDIX

A GIBBS SAMPLING FOR NRTG

Exact posterior inference is intractable in our model, therefore, we turn to a collapsed Gibbs sampling algorithm for approximate posterior inference, which is simple to derive, comparable in speed to other estimators, and can approximate a global maximum [8].

Integrating out $\boldsymbol{\vartheta}$ and $\boldsymbol{\varphi}$ analytically, the latent variable needed by the sampling algorithm is the pattern indicator r. Dirichlet hyperparameters α and β are fixed before sampling. The parameter ψ of the continuous Beta distribution is updated after each Gibbs sample by the method of moments. Due to the spatio-temporal neighbor regularization, we also have ξ to be estimated during sampling (see Section 4.2).

Sampling pattern assignments $\boldsymbol{r}$. Given rest variables, sampling r is similar to sampling approach for Topics over Time [36]. The difference lies in that RTG has a spatio-temporal neighbor regularization. That is,

$$\begin{aligned} p(r_{mn} = r) = \left(n_m^{(r)} + \alpha^{(r)} - 1\right) \frac{n_r^{(v)} + \beta^{(v)} - 1}{\sum_{v=1}^{V}(n_r^{(v)} + \beta^{(v)}) - 1} \\ \times \frac{(1 - c_{mn})^{\psi_\alpha - 1} c_{mn}^{\psi_\beta - 1}}{B(\psi_\alpha, \psi_\beta)} \exp\left(\frac{-(\xi_v^{(r)} - \varphi_v^{(r)})^2}{2\sigma_\xi^2}\right), \end{aligned} \tag{26}$$

where $n_m^{(r)}$ indicates the number of ST-Zones assigned to pattern r in the mth trajectory, and $n_r^{(v)}$ indicates the number of times ST-Zone v is assigned to pattern r.

Updating of $\boldsymbol{\xi}$. With the sampled r, we can obtain the estimated $\boldsymbol{\varphi}$ (see Section 4.2), which enables us to estimate the ξ (see Section 4.2).

Updating of $\boldsymbol{\psi}$. For simplicity, we update ψ after each Gibbs sample by the method of moments, detailed as follows:

$$\hat{\psi}_\alpha = \bar{c}_r \left(\frac{\bar{c}_r(1 - \bar{c}_r)}{b_r^2} - 1\right), \ \hat{\psi}_\beta = (1 - \bar{c}_r)\left(\frac{\bar{c}_r(1 - \bar{c}_r)}{b_r^2} - 1\right), \tag{27}$$

where $\bar{c}_r$ and b_r indicate, respectively, the sample mean and the biased sample variance of the crowd scores c belonging to pattern r.

B COMPLEXITIES OF NRTG

We first discuss the time complexity of the nRTG model. As shown in the APPENDIX A, the major computation cost during the Gibbs sampling of nRTG is inferring value of latent variable $r_{m,n}$. Specifically, for each zone, we need to infer $r_{m,n}$ takes which value in the set $\{1, \ldots, K\}$, i.e., the probability $p(r_{m,n} = k|\text{rest})$ for each $k \in \{1, \ldots, K\}$ need to be computed. Thus, the time complexity of sampling the risk pattern for each zone is $O(K)$. Therefore, for the V zones in the ST-Cube, the time complexity of the Gibbs sampling is $O(VK)$.

Next, we analyze the space complexity. For the nRTG model, the parameters include $\{\psi^{(k)}\}_{k=1}^K$, $\{\phi^{(k)}\}_{k=1}^K$, $\{\theta^{(m)}\}_{m=1}^M$, and $\{\hat{\xi}_v\}_{v=1}^V$, where K is the number of patterns, M is the number of trajectories, and V is the number of zones. As each $\psi^{(k)}$ has two dimensions, $\{\psi^{(k)}\}_{k=1}^K$ has $2K$ parameters. Similarly, each $\phi^{(k)}$ has V dimensions, $\{\phi^{(k)}\}_{k=1}^K$ has VK parameters. Each $\theta^{(m)}$ has K dimensions, therefore, $\{\theta^{(m)}\}_{m=1}^M$ has MK parameters. $\{\hat{\xi}_v\}_{v=1}^V$ has V parameters. In summary, the space complexity of nRTG is $O(VK + 2K + MK + V) = O((V + 2 + M)K + V)$.

REFERENCES

[1] Usman Ali and Tariq Mahmood. 2017. Using deep learning to predict short term traffic flow: A systematic literature review. In *Proceedings of the 1st International Conference on Intelligent Transport Systems*. Springer, 90–101.

[2] Derya Birant and Alp Kut. 2007. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data Knowl. Eng.* 60, 1 (2007), 208–221.

[3] Sanjay Chawla, Yu Zheng, and Jiafeng Hu. 2012. Inferring the root cause in road traffic anomalies. In *Proceedings of the IEEE 12th International Conference on Data Mining (ICDM'12)*. IEEE, 141–150.

[4] Hongmei Chen, Yixiang Fang, Ying Zhang, Wenjie Zhang, and Lizhen Wang. 2019. ESPM: Efficient spatial pattern matching. *IEEE Trans. Knowl. Data Eng.* 32, 6 (2019), 1227–1233.

[5] Ghyzlane Cherradi, Adil EL Bouziri, Azedine Boulmakoul, and Karine Zeitouni. 2017. Real-time microservices based environmental sensors system for Hazmat transportation networks monitoring. *Transport. Res. Procedia* 27 (2017), 873–880.

[6] A. Ditta, O. Figueroa, G. Galindo, and R. Yie-Pinedo. 2018. A review on research in transportation of hazardous materials. *Socio-Econ. Plan. Sci.* 68, 12 (2018), 100665.

[7] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Conference on Knowledge Discovery and Data Mining*, Vol. 96. 226–231.

[8] Edward I. George and Robert E. McCulloch. 1993. Variable selection via Gibbs sampling. *J. Amer. Statist. Assoc.* 88, 423 (1993), 881–889.

[9] Tony H. Grubesic, Ran Wei, and Alan T. Murray. 2014. Spatial clustering overview and comparison: Accuracy, sensitivity, and computational expense. *Ann. Assoc. Amer. Geog.* 104, 6 (2014), 1134–1156.

[10] Samiul Hasan and Satish V. Ukkusuri. 2014. Urban activity pattern classification using topic models from online geo-location data. *Transport. Res. Part C: Emerg. Technol.* 44 (2014), 363–381.

[11] Gregor Heinrich. 2005. *Parameter Estimation for Text Analysis.* Technical Report. Fraunhofer Institute for Computer Graphics Research.

[12] Henry Hsu and Peter A. Lachenbruch. 2005. Paired t test. *Encycl. Biostat.* 6 (2005).

[13] Bo Hu, Mohsen Jamali, and Martin Ester. 2013. Spatio-temporal topic modeling in mobile social media for location recommendation. In *Proceedings of the IEEE 13th International Conference on Data Mining*. IEEE, 1073–1078.

[14] Xifei Huang, Xinhao Wang, Jingjing Pei, Ming Xu, Xiaowu Huang, and Yun Luo. 2018. Risk assessment of the areas along the highway due to hazardous material transportation accidents. *Nat. Haz.* 93, 3 (2018), 1181–1202.

[15] Yan Huang, Shashi Shekhar, and Hui Xiong. 2004. Discovering colocation patterns from spatial data sets: A general approach. *IEEE Trans. Knowl. Data Eng.* 16, 12 (2004), 1472–1485.

[16] Yan Huang, Liqin Zhang, and Pusheng Zhang. 2008. A framework for mining sequential patterns from spatio-temporal event data sets. *IEEE Trans. Knowl. Data Eng.* 20, 4 (2008), 433–448.

[17] Renhe Jiang, Xuan Song, Dou Huang, Xiaoya Song, Tianqi Xia, Zekun Cai, Zhaonan Wang, Kyoung-Sook Kim, and Ryosuke Shibasaki. 2019. DeepUrbanEvent: A system for predicting citywide crowd dynamics at big events. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2114–2122.

[18] Masoud Khanmohamadi, Morteza Bagheri, Navid Khademi, and Seyed Farid Ghannadpour. 2018. A security vulnerability analysis model for dangerous goods transportation by rail–Case study: Chlorine transportation in Texas-Illinois. *Safety Sci.* 110 (2018), 230–241.

[19] Yaguang Li and Cyrus Shahabi. 2018. A brief overview of machine learning methods for short-term traffic forecasting and future directions. *SIGSPATIAL Special* 10, 1 (2018), 3–9.

[20] Yexin Li and Yu Zheng. 2019. Citywide bike usage prediction in a bike-sharing system. *IEEE Trans. Knowl. Data Eng.* 32, 6 (2019), 1079–1091.

[21] Wei Liu, Yu Zheng, Sanjay Chawla, Jing Yuan, and Xie Xing. 2011. Discovering spatio-temporal causal interactions in traffic data streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1010–1018.

[22] Yu Liu, Martin Ester, Bo Hu, and David W. Cheung. 2015. Spatio-temporal topic models for check-in data. In *Proceedings of the IEEE International Conference on Data Mining*. IEEE, 889–894.

[23] Top News. 2016. Guijie comprehensive transformation of start tomorrow. Retrieved from http://www.top-news.top/news-12386939.html.

[24] Mohammad Noureddine and Milos Ristic. 2019. Route planning for hazardous materials transportation: Multicriteria decision making approach. *Dec. Mak.: Applic. Manag. Eng.* 2, 1 (2019), 66–85.

[25] E. Planas, E. Pastor, F. Presutto, and J. Tixier. 2008. Results of the MITRA project: Monitoring and intervention for the transportation of dangerous goods. *J. Haz. Mat.* 152, 2 (2008), 516–526.

[26] Francisco Enrique Santarremigia, Gemma Dolores Molero, Sara Poveda-Reyes, and José Aguilar-Herrando. 2018. Railway safety by designing the layout of inland terminals with dangerous goods connected with the rail transport system. *Safety Sci.* 110 (2018), 206–216.

[27] Xiaoying Shi, Fanshun Lv, Dewen Seng, Baixi Xing, and Jing Chen. 2019. Exploring the evolutionary patterns of urban activity areas based on origin-destination data. *IEEE Access* 7 (2019), 20416–20431.

[28] Ying Sun, Hengshu Zhu, Fuzhen Zhuang, Jingjing Gu, and Qing He. 2018. Exploring the urban region-of-interest through the analysis of online map search queries. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2269–2278.

[29] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM'06)*. IEEE, 613–622.

[30] Jingyuan Wang, Chao Chen, Junjie Wu, and Zhang Xiong. 2017. No longer sleeping with a bomb: A duet system for protecting urban safety from dangerous goods. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1673–1681.

[31] Jingyuan Wang, Qian Gu, Junjie Wu, Guannan Liu, and Zhang Xiong. 2016. Traffic speed prediction and congestion source exploration: A deep learning method. In *Proceedings of the IEEE 16th International Conference on Data Mining (ICDM'16)*. IEEE, 499–508.

[32] Jingyuan Wang, Xiaoda Wang, Chao Li, Junjie Wu, et al. 2020. Deep fuzzy cognitive maps for interpretable multivariate time series prediction. *IEEE Trans. Fuzzy Syst.* (2020).

[33] Jingyuan Wang, Junjie Wu, Ze Wang, Fei Gao, and Zhang Xiong. 2019. Understanding urban dynamics via context-aware tensor factorization with neighboring regularization. *IEEE Trans. Knowl. Data Eng.* 32, 11 (2019), 2269–2283.

[34] Jingyuan Wang, Ning Wu, Xinxi Lu, Xin Zhao, and Kai Feng. 2019. Deep trajectory recovery with fine-grained calibration using Kalman filter. *IEEE Trans. Knowl. Data Eng.* 33, 3 (2019), 921–934.

[35] Jingyuan Wang, Ning Wu, Wayne Xin Zhao, Fanzhang Peng, and Xin Lin. 2019. Empowering A* search algorithms with neural networks for personalized route recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 539–547.

[36] Xuerui Wang and Andrew McCallum. 2006. Topics over time: A non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 424–433.

[37] Yuan Wang, Dongxiang Zhang, Ying Liu, Bo Dai, and Loo Hay Lee. 2018. Enhancing transportation systems via deep learning: A survey. *Transport. Res. Part C: Emerg. Technol.* 99, 2 (2018), 144–163.

[38] Yingzi Wang, Xiao Zhou, Anastasios Noulas, Cecilia Mascolo, Xing Xie, and Enhong Chen. 2018. Predicting the spatio-temporal evolution of chronic diseases in population with human mobility data. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 3578–3584.

[39] Ning Wu, Xin Wayne Zhao, Jingyuan Wang, and Dayan Pan. 2020. Learning effective road network representation with hierarchical graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 6–14.

[40] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 186–194.

[41] Nicholas Jing Yuan, Yu Zheng, Xing Xie, Yingzi Wang, Kai Zheng, and Hui Xiong. 2015. Discovering urban functional zones using latent activity trajectories. *IEEE Trans. Knowl. Data Eng.* 27, 3 (2015), 712–725.

[42] Huichu Zhang, Yu Zheng, and Yong Yu. 2018. Detecting urban anomalies using multiple spatio-temporal data sources. *Proc. ACM Interact., Mob., Wear. Ubiq. Technol.* 2, 1 (2018), 54.

[43] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.* 5, 3 (2014), 38.

[44] Xiao Zhou, Anastasios Noulas, Cecilia Mascolo, and Zhongxiang Zhao. 2018. Discovering latent patterns of urban cultural interactions in WeChat for modern city planning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1069–1078.

[45] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning (ICML'03)*. 912–919.

Received October 2020; revised January 2021; accepted January 2021