Precision CityShield Against Hazardous Chemicals Threats via Location Mining and Self-Supervised Learning

Jiahao Ji[*]
Jingyuan Wang[†]
School of Computer Science and Engineering, Beihang University
Beijing, China
Pengcheng Laboratory
Shenzhen, China

Junjie Wu
School of Economics and Management, Beihang University
Beijing, China
Beijing Key Laboratory of Emergency Support Simulation Technologies for City Operations, Beihang University
Beijing, China

Boyang Han
Junbo Zhang
Yu Zheng
JD Intelligent Cities Research
Beijing, China
JD iCity, JD Technology
Beijing, China

ABSTRACT

With the unprecedented development of industrialization and urbanization, many hazardous chemicals have become an indispensable part of our daily life. They are produced, transported, and consumed in modern cities every day, which breeds many unknown hazardous chemicals-related locations (HCLs) that are out of the supervision of management departments and accompanying huge threats to urban safety. How to recognize these unknown HCLs and identify their risk levels is an essential task for urban hazardous chemicals management. To accomplish this task, in this work, we propose a system named as CityShield to discover hidden HCLs and classify their risk levels based on trajectories of hazardous chemicals transportation vehicles. The CityShield system consists of three components. The first component is *Data Pre-processing*, which filters noises in raw trajectories and probes stable transportation vehicles' stay points from massive uncertain GPS points. The second is *HCL Recognition*, which adopts the proposed HCL-Rec algorithm to cluster stay points into polygonal HCLs, and avoids the improper location merging problem caused by the skewed spatial distribution of HCLs. The third component is *HCL Classification*, which introduces the HCL relation graph as auxiliary information to overcome the label scarcity problem of HCLs. It adopts a self-supervised method consisting of four pre-training tasks to learn high-quality representations for HCLs from the graph, which are finally used to classify the categories and risk levels of HCLs.

The CityShield system has been deployed in Nantong, an important hazardous chemicals import and export city in China. Experiments and case studies on two large-scale real-world datasets collected from Nantong demonstrated the effectiveness of the proposed system. In real-world applications, the CityShield system discovered 173 high-risk unknown HCLs for the Nantong government, and successfully moved the hazardous chemicals management of Nantong to the prevention rather than emergency response side.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; • **Applied computing** → *Computers in other domains.*

KEYWORDS

Hazardous chemicals management, Spatio-temporal data mining, Representation learning, Urban computing

ACM Reference Format:
Jiahao Ji, Jingyuan Wang, Junjie Wu, Boyang Han, Junbo Zhang, and Yu Zheng. 2022. Precision CityShield Against Hazardous Chemicals Threats via Location Mining and Self-Supervised Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539028

1 INTRODUCTION

Hazardous chemicals have been widely used and become an indispensable part of our daily life, such as acid for battery, ammonium nitrate for fertilizer, liquefied gases for cooking, *etc.* Due to the very nature of being flammable, explosive and toxic, however, hazardous chemicals need to be carefully regulated during their entire lifecycle, otherwise they can pose fatal risks to urban public safety. For example, on August 4, 2020, a hazardous chemicals-related location (HCL) storing ammonium nitrate at the Port of Beirut exploded (see Fig. 1(a)). This accident caused more than 218 deaths, 7,000 injuries, and $15 billion in property damage, and left estimated 300,000 people homeless[1]. While improving urban emergency management is important to deal with this kind of accidents, the ability to scent the risks and move to the prevention side is the fundamental solution. If the high-risk HCLs are identified in advance, authorities can best deploy their limited resources to prevent such tragedies and ensure urban safety.

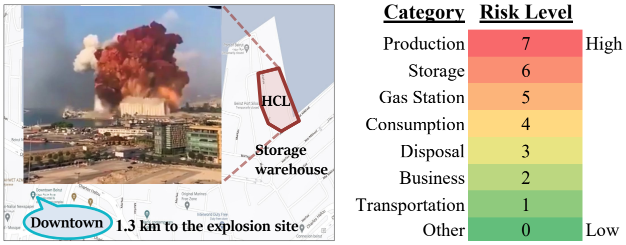
In the literature, studies related to hazardous chemicals risk management usually focus on risk analysis [27] or route planning [11] during transportation process. Most of these works are based on the assumption that all HCLs are known by the management departments. However, in practice, such as indicated by the real-world dataset used in this work, more than 90% HCLs could be unknown

[*] The paper was done when Jiahao Ji was an intern at JD Intelligent Cities Research under the supervision of the Junbo Zhang (msjunbozhang@outlook.com).
[†] Corresponding author: jywang@buaa.edu.cn

[1] https://en.wikipedia.org/wiki/2020_Beirut_explosion

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539028



(a) Beirut port explosion (b) Location risk level

Figure 1: (a) shows the explosion on August 4, 2020. (b) gives the location risk level and the corresponding category.

to the authorities, which causes huge hazardous chemicals related risks to cities. How to recognize these hidden HCLs and identify their risk levels is an essential task for urban safety management. Fortunately, with the advance in big data technology, the trajectory data of hazardous chemicals transportation vehicles are becoming very easy to obtain, from which we can identify the driving routes and parking locations of hazardous chemicals transportation vehicles, and intuitively a considerable part of these locations are HCLs. Transportation trajectories of hazardous chemicals also provide rich semantic information, from which we can infer the risk levels and categories of hidden HCLs.

Based on this insight, in this paper, we propose CityShield, an HCL recognizing and risk level identifying system based on hazardous chemicals transportation trajectories. We design three components in the CityShield system, *i.e.,* Data Pre-processing, HCL Recognition and HCL Classification components, to handle three challenges in mining HCLs from complex hazardous chemicals transportation trajectories. The first challenge is that trajectories of transportation vehicles are typically full of noises and uncertainties. To extract stable location patterns from these noisy and uncertain trajectory data is essentially non-trivial. To deal with this, the Data Pre-processing component in the CityShield system adopts a noise filter to reduce random noises in trajectories and unveil transportation vehicles' stable stay points from massive uncertain GPS locations.

The second challenge comes from the highly skewed spatial distribution of HCLs, *i.e.,* a small region of a city may contain a lot of HCLs (for example, in our experimental dataset, over 80% of HCLs locate in several small chemical industry parks). This characteristic of HCLs brings great challenge to recognizing individual HCLs accurately, since it is very easy to merge several neighboring HCLs into one HCL. To overcome this problem, we design an HCL-Rec algorithm in the HCL Recognition component. Based on the insight that different HCLs usually employ different transportation vehicles, the algorithm recognizes HCLs through clustering vehicles' stay points with their ID information, which helps to elimiate the HCLs merging issue.

The third and biggest challenge is for HCL risk level and category identifying. Sine the majority of HCLs are unknown to the authorities, we have very scarce category and risk level labels of HCLs, which prevents traditional supervised classification methods from being directly used. Inspired by recent advances in self-supervised pre-training methods, we design a pre-training-based HCL representation learning method for the HCL Classification component of the CityShield system. Specifically, we first introduce the global HCL graph to model long-term HCL relations in trajectories, and design two predictive pre-training tasks including degree and context predictions to exploit the useful information in the HCL graph. Next, we construct local HCL graphs that record short-term HCL relations as a data augmentation of the global HCL graph, and propose two contrastive pre-training tasks including local-local and local-global contrasts to ensure the consistency of HCL representations in both local and global graphs. Finally, the discriminative HCL representations that are learned by the pre-training mechanism are used in the final HCL risk level and category classification.

As a complete solution, CityShield has been deployed by an important hazardous chemicals import and export city: Nantong, China[2]. Extensive experiments as well as case studies on two real-world datasets from Nantong show the effectiveness of the proposed methods in CityShield. The deployment of the CityShield system has also achieved great practical effects. Since September 2021, the system has helped the authorities to identify 173 hidden high-risk HCLs. The ability of urban hazardous chemicals management of Nantong has been significantly improved by our CityShield system, especially on the prevention rather than emergence response side.

2 OVERVIEW

This section gives the basic concepts and introduces the overall framework of CityShield.

2.1 Preliminaries

Definition 1 (Trajectory). *A trajectory is a sequence of location points generated by a vehicle, denoted as $tr : p_1 \rightarrow \cdots \rightarrow p_m$, where each point $p = (vid, lng, lat, t)$, vid is the vehicle ID, (lng, lat)* (a.k.a., *longitude and latitude) indicates the location, and t is the timestamp. The set of trajectories is denoted as $\mathcal{T}$.*

All trajectories used in the CityShield system are collected from hazardous chemicals transportation vehicles. These vehicles are mandatorily equipped with GPS devices to report real-time locations to the local government, which are then aggregated into the trajectory dataset $\mathcal{T}$.

Definition 2 (Stay Point). *A stay point of a vehicle, denoted as $sp = (vid, lng, lat, tp)$, stands for a geographic region centered on (lng, lat), at which a vehicle vid stays for a period of time tp* (e.g., *30 minutes). The set of stay points is denoted as $\mathcal{S}$.*

Compared with raw GPS points, a stay point contains particular semantic information, such as the place of vehicles loading/unloading hazardous chemicals.

Definition 3 (Hazardous Chemicals-related Location, HCL). *An HCL is a geographic region that contains various correlated stay points of hazardous chemicals transportation vehicles. Its boundary is generated by a group of stay points $\{sp_1, \ldots, sp_M\}$. The set of HCLs is denoted as $\mathcal{N} = \{n_1, \ldots, n_i, \ldots, n_I\}$, where n_i denotes a HCL.*

Definition 4 (HCL Risk Level). *In the CityShield system, each HCL is assigned with a Risk Level based on the category of its function,*

[2] https://en.wikipedia.org/wiki/Nantong

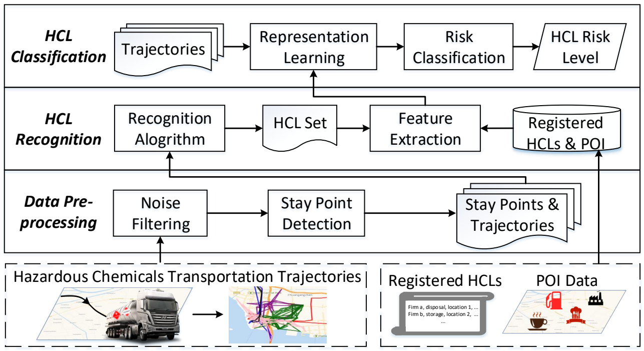


Figure 2: System framework of CityShield.

ranging from 0-7 (see Fig. 1(b)). The higher the level, the greater the risk.

2.2 System Framework

Our CityShield system has two main functions: *i*) recognizing HCLs from massive trajectories, and *ii*) assigning HCLs with proper risk levels. The overall framework of CityShield is elaborated in Fig. 2, which mainly consists of three components as follows.

Data Pre-processing. This component takes hazardous chemicals transportation trajectories $\mathcal{T}$ as input and outputs a stay point set $\mathcal{S}$. There are two main tasks: *i*) *Noise Filtering*, which cleans trajectories by removing outlier GPS points; *ii*) *Stay Point Detection*, which detects stay points $\mathcal{S}$ from the filtered trajectories. Section 3 gives the details.

HCL Recognition. This component takes the stay point set $\mathcal{S}$ as input and generates HCLs with their boundaries and features. It includes two steps: *i*) *HCLs Recognition*, which recognizes all HCLs, denoted as the set $\mathcal{N}$, by using the stay points generated by the data pre-processing component; *ii*) *Feature Extraction*, which extracts static and dynamic features for each HCL by making use of POI data and stay points, respectively. Section 4 gives the details.

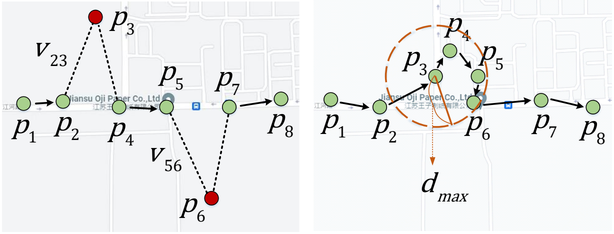
HCL Classification. This component takes the HCL set $\mathcal{N}$ as input and classifies the HCLs into different risk levels. It includes two main steps: *i*) *Representation Learning*, which learns representations for each HCL through building a HCL relation graph and designing four self-supervised graph representation learning tasks; *ii*) *Risk Classification*, which classifies HCLs into different risk levels based on the learned representations. Section 5 gives the details.

3 DATA PRE-PROCESSING

In this section, we introduce the data pre-processing component of the CityShield system.

3.1 Noise Filtering

The hazardous chemicals transportation trajectory data generated by GPS are never perfectly accurate due to varied atmospheric conditions and signal blockage. Fig. 3(a) shows a trajectory with noise GPS points, where points p_3 and p_6 might be several hundred meters away from their true locations. Such noise GPS points would affect the quality of stay point detection and the subsequent tasks. We therefore adopt a heuristic approach proposed in [25] to automatically filter noise GPS points in trajectories.



(a) Noise filtering. (b) Stay point detection.

Figure 3: Illustrations of data pre-processing.

Concretely, the algorithm sequentially calculates the traveling speed for each point in a trajectory based on its precursor point and itself. If the calculated speed is larger than a threshold, the corresponding point is deemed as noise and should be removed from the trajectory. For example, in Fig. 3(a), because the speeds v_{23} and v_{56} are larger than the speed threshold, the corresponding points p_3 and p_6 are removed from the trajectory.

3.2 Stay Point Detection

The function of this component is to acquire stay points from the filtered trajectories. Based on the trajectories after noise filtering, we adopt the stay point detection algorithm proposed in [10] to extract stay points.

The algorithm traverses points over a trajectory to check whether the distance between an anchor GPS point and its successor points is shorter than a threshold d_{max}. For the successor points with the distances lower than d_{max}, the algorithm further checks whether the duration between the anchor point and the last successor point is larger than a time threshold t_{min}. As shown in Fig. 3(b), we assume p_3 is the current anchor point, and $\{p_4, p_5, p_6\}$ are its only successors within distance threshold d_{max}. The algorithm then calculates the time interval between the anchor point p_3 and its last successor p_6 within d_{max}. If the duration is larger than a time threshold t_{min}, *i.e.*, $p_6.t - p_3.t > t_{min}$, the algorithm generates the center coordinates of a stay point as

$$sp.lng = \frac{\sum_{k=3}^{6} p_k.lng}{4}, \quad sp.lat = \frac{\sum_{k=3}^{6} p_k.lat}{4}, \tag{1}$$

and generates the timestamp of the stay point as

$$sp.tp = p_3.t + \frac{(p_6.t - p_3.t)}{2}. \tag{2}$$

Then, the algorithm moves the anchor point to the next GPS location after the current stay point, *i.e.*, p_7. The entire process is repeated until the anchor point moves to the end of a raw trajectory.

After data pre-processing, we obtain a stay point set $\mathcal{S}$ in which each vehicle's stay points are organized together for HCL recognition.

4 HCL RECOGNITION

The function of this component is to recognize HCLs from stay point set $\mathcal{S}$ and extract features of HCLs for risk classification.

Algorithm 1 HCL-Rec

```
Input: Stay point set $\mathcal{S}$, recognition radius $r$.
Output: HCL set $\mathcal{N}$.
1:  Select an unvisited seed $s \in \mathcal{S}$.
2:  Initialize set $Q_l = \{s\}$ and queue $R = [\ ]$.
3:  while not $Q_l.empty()$ do
4:      for $a \in Q_l$ do                                  ▷ Iteration starts.
5:          Let queue $Q = [s]$.
6:          while not $Q.empty()$ do
7:              $q = Q.pop(), R.push(q)$.
8:              for $p \in \mathcal{S}$ do                  ▷ Ensure $p$ is unvisited
9:                  if $(p.vid == q.vid) \wedge (distance(p, q) \leq r)$ then
10:                     $Q.push(p)$.                        ▷ Iteration ends.
11:     Yield bounding polygon $n$ using stay points in $R$.
12:     Refill $Q_n$ with unvisited stay points within polygon $n$.
13: $\mathcal{N}.add(n)$.
14: if there exists unvisited stay points in $\mathcal{S}$ then
15:     Repeat step 1 to 13.
16: Merge the overlapping HCLs in $\mathcal{N}$.
17: return $\mathcal{N}$.
```

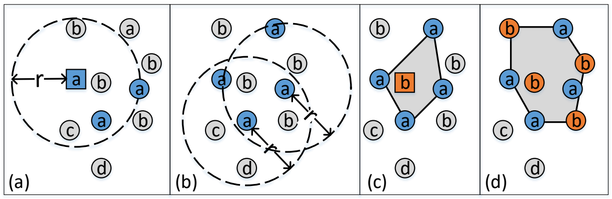


Figure 4: Illustration of HCL-Rec. Markers with content (*i.e.,* vehicle ID) denote stay points. r is the recognition radius.

4.1 Recognition Algorithm

According to Definition 3, an HCL is a geographic region containing stay points of various hazardous chemicals transportation vehicles. Therefore, a straight-forward method to recognize HCLs from stay point set is to use density clustering algorithms, such as DBSCAN [1]. However, in real-world practices, the spatial distribution of stay points is typically skewed, *i.e.,* many stay points are located in small regions, traditional density clustering algorithms are prone to mistakenly merging several HCLs into one.

To address this issue, we propose an HCL recognition algorithm named HCL-Rec, which is based on the insight that different chemical factories generally hire different vehicles to transport hazardous chemicals. HCL-Rec is a density-based clustering algorithm that incorporates vehicle ID information. Specifically, HCL-Rec iteratively clusters stay points of each vehicle in a potential HCL. For each iteration, the algorithm only focuses on one vehicle's stay points, regardless of how close other vehicles' stay points are to the current cluster center. This strategy avoids gathering stay points of different HCLs into one cluster.

Alg. 1 gives the pseudocodes of the proposed HCL-Rec algorithm. Note that the recognition radius r as input is the distance threshold for determining whether a stay point belongs to the current HCL. To facilitate understanding, we provide an illustration in Fig. 4, where we use a, b, c to distinguish the vehicle IDs to which the stay points belong.

Fig. 4(a): In Lines 1-2, the algorithm initializes all stay points as unvisited and sets vehicle a's stay point in the square marker as the seed stay point. r is the input recognition radius.

Fig. 4(b): In Lines 3-10, starting with the seed stay point a of square, the algorithm iteratively selects stay points with the same $vid = a$ inside the recognition radius r and labels the selected stay points as visited, until there exists no unvisited stay points of vehicle a within the recognition radius r.

Fig. 4(c): In Line 11, the algorithm yields a polygon as a potential HCL n' using the visited stay points. In Line 12, the algorithm selects an unvisited stay point within n', *i.e.,* b in the square marker belonging to a new vehicle, as a new seed for the next iteration.

Fig. 4(d): The algorithm repeats steps (b) and (c) (*i.e.,* Lines 3-12) to expand the current potential HCL n' until there exists no unvisited stay point within n'.

Finally, we merge the overlapped HCLs to eliminate the randomness of HCL-Rec in Line 16. In this way, the HCL-Rec algorithm generates the same final HCL set no matter how we start the algorithm to form HCLs, from vehicle a or b.

4.2 Feature Extraction

In this component, we extract features for each HCL. The extracted features $\boldsymbol{x}_i \in \mathbb{R}^F$ describe HCL n_i in combination with the static features and dynamic features.

- **Static features**: *i*) the count and frequency of POIs inside each HCL; *ii*) the statistical information of each HCL, including area, polyline number of its boundary, and Geohash[3] number inside it.
- **Dynamic features**: the count and frequency of the stay duration, arrival/departure time of day (48 time slices per day), arrival day of week, and vehicle ownership of stay points inside each HCL. The ownership is divided into three categories: the city, other cities in the province, and other provinces.

The output of the HCL recognition component is an HCL set denoted as $\mathcal{N} = \{n_1, \ldots, n_N\}$ with their features. In urban management, these HCLs need to be closely monitored. Our CityShield system provides an effective tool to discover these risky areas for urban safety management.

5 HCL CLASSIFICATION

In the HCL set $\mathcal{N}$, some HCLs are known to and supervised by authorities, but some are unknown before and are discovered by our proposed algorithm. These unknown HCLs are not under the supervision of authorities and thus form the main source of hidden dangers. For the unknown HCLs, we need to further assign categories and risk levels to them so as to help authorities better manage high-risk HCLs. As shown in Fig. 1(b), we classify HCLs into eight categories, which are Production, Storage, Gas Station, Consumption, Disposal, Business, Transportation, and Other. The risk level decreases from the first category to the last one. In this section, we introduce the HCL category classification component of the CityShield system.

5.1 Model Framework

In the HCL classification component, we adopt a supervised learning framework to assign risk level to each HCL, *i.e.,* we use a training

[3] https://en.wikipedia.org/wiki/Geohash

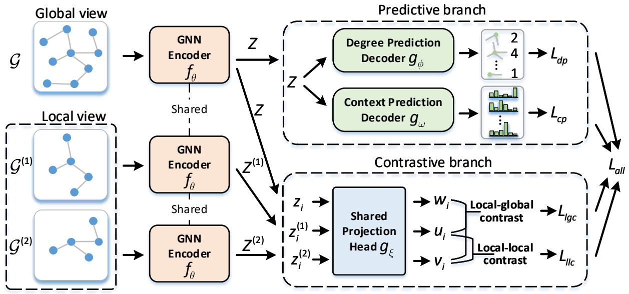


Figure 5: Structure of representation learning model.

set consisting of HCLs with risk level labels to train a classifier, and then use the classifier to assign risk level to the HCLs without labels. As previously mentioned, some HCLs are supervised by authorities, and the risk levels of these HCLs are known to us, which can serve as the training set with labels. Nevertheless, in real-world applications, the ratio of the labeled HCLs is very low, which will degrade the classification performance seriously. One possible way to alleviate this problem is to explore more information from the unlabeled data.

Along this line, we develop a self-supervised representation learning model to generate high-quality representations for HCLs, and then use the representations to classify HCLs. Specifically, we build an HCL graph using relations among HCLs, and adopt a Graph Neural Network (GNN) encoder to generate representations for each HCL. The HCL graph describes the HCL relations in hazardous chemicals transportation trajectories, which is defined as follows.

Definition 5 (HCL Graph). *An HCL graph is denoted as* $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \boldsymbol{X})$*, where the HCL set* $\mathcal{N} = \{n_1, \ldots, n_N\}$ *is the node set,* $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ *is the edge set modelling the relations among the HCLs, and* $\boldsymbol{X} \in \mathbb{R}^{N \times F}$ *is the feature matrix where* $\boldsymbol{x}_i \in \mathbb{R}^F$ *is the feature vector of the HCL* n_i*. The adjacency matrix of* $\mathcal{G}$ *is* $\boldsymbol{A} \in \{0,1\}^{N \times N}$ *with* $a_{ij} = 1$ *if there exists a trajectory from* n_i *to* n_j (i.e., $e_{ij} \in \mathcal{E}$) *and* $a_{ij} = 0$ *otherwise.*

We construct a global HCL graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}, \boldsymbol{X}\}$ using all trajectories. Based on $\mathcal{G}$, we use a GNN encoder f_θ to generate the representation vectors of HCLs as

$$Z = f_\theta(\mathcal{G}), \tag{3}$$

where $\boldsymbol{Z} \in \mathbb{R}^{N \times D}$ denotes the representation matrix, and the i-th row vector, *i.e.,* $\boldsymbol{z}_i \in \mathbb{R}^D$, is the representation vector of HCL n_i. The GNN encoder f_θ is implemented by a two-layer graph convolutional network [9].

5.2 HCL Graph Representation Learning

As shown in Fig. 5, we adopt two types of pre-training tasks to train the representation generator f_θ. One is predictive task and the other is contrastive learning task. Each type contains two specific tasks. We introduce these tasks as follows.

5.2.1 Predictive Task. The predictive tasks adopted by our model contains *Degree Prediction* and *Context Prediction*, both of which use HCL representations to predict discriminative features of HCL graph nodes. We use these tasks to exploit useful characteristics of the HCL graph. The discriminative features include the degree of nodes and the neighbor context.

Predictive Task #1: Degree Prediction. In the HCL graph, the degree distribution of nodes is a discriminative feature for different HCL categories. For example, the degree distribution of hazardous chemicals storages is significantly different from that of gas stations. Because a hazardous chemicals storage transports gasoline to multiple gas stations, while a gas station usually gets gasoline from a fixed storage. As shown in Fig. 1(b), the risk level of storage is also higher than that of gas stations. If a model can predict HCL degrees, the representations learned by the model could be more helpful to the HCL categorization.

Given the representation $\boldsymbol{z}_i$ of any HCL $n_i \in \mathcal{N}$ generated by the GNN encoder f_θ, the objective of *degree prediction* task is to minimize the mean square error (MSE) loss between real and predicted degree of n_i, *i.e.,*

$$L_{dp} = \frac{1}{|\mathcal{N}|} \sum_{n_i \in \mathcal{N}} \left(g_\phi\left(\boldsymbol{z}_i\right) - d_i\right)^2, \tag{4}$$

where $g_\phi(\cdot)$ is the degree predictor implemented by a linear regression, and d_i is the degree of HCL n_i. Based on the HCL graph topology, we calculate the degree of each HCL n_i by $d_i = \sum_j a_{ji} + \sum_j a_{ij}$, where a_{ij} is the element of adjacent matrix $\boldsymbol{A}$ for the HCL graph $\mathcal{G}$.

Predictive Task #2: Context Prediction. This task is designed to exploit the discriminative features hidden in neighbor nodes of an HCL. For HCLs in different categories, their neighbor nodes in the HCL graph are usually different. Sometimes, neighbors are even more discriminative than the raw features of HCLs extracted in Section 4.2. For example, in a chemical industrial park, oil refineries (production category) and storage warehouses (storage category) may be spatially close, leading to similar features of them. In the HCL graph, however, oil refineries are mostly neighbors of gas stations, while storage warehouses are neighbors of various production factories. To utilize such neighbor context information for better HCL representation learning, we introduce a *context prediction* task.

We first construct a context vector $\boldsymbol{c}_i$ for HCL n_i. Let K be the number of HCL categories. The k-th element of $\boldsymbol{c}_i$ is defined as:

$$c_{ik} = \frac{B_{i,k}}{B_i}, \ k = 1, \ldots, K, \tag{5}$$

where B_i denotes the number of n_i's neighbors, and $B_{i,k}$ denotes the number of neighbors belonging to category k. Based on the context label vector $\boldsymbol{c}_i$, we employ the MSE optimization objective as follows:

$$L_{cp} = \frac{1}{|\mathcal{N}|} \sum_{n_i \in \mathcal{N}} \|g_\omega\left(\boldsymbol{z}_i\right) - \boldsymbol{c}_i\|^2, \tag{6}$$

where $g_\omega(\cdot)$ is the context predictor implemented by a SoftMax regression.

To ensure all neighbors have categories in the calculation of $B_{i,k}$, we adopt the label propagation algorithm [28] to assign a category to each HCL.

5.2.2 Contrastive Task. The contrastive task aims to improve representation quality using a data augmentation method. Traditional data augmentation in graph representation learning is usually through random perturbing edges or node attributes [29]. This approach may cause loss of information and introduce unnecessary noise. We overcome this shortage by introducing a local HCL graph concept.

Specifically, a local HCL graph records short-term HCL relations, which is constructed by trajectories of a short period of time, *e.g.,* one day. Since the global HCL graph[4] records all HCL relations, the local HCL graph is a kind of sub-graph of the global HCL graph. From the trajectory data, we can generate many local HCL graphs as augmented data of the global HCL. Compared with the random perturbation approaches, local HCL graphs have clear physical meanings.

Since the categories of HCLs are invariant in both local and global graphs, the HCLs representations between local and global graphs should be consistent. Based on this insight, we propose two contrastive tasks, *i.e.,* local-local contrast and local-global contrast, to learn the representations of HCLs.

Contrastive Task #1: Local-Local Contrast. The *local-local contrast* task ensures that the representations between different local graphs are consistent. Specifically, given two local graphs $\mathcal{G}^{(1)} = \{\mathcal{N}^{(1)}, \mathcal{E}^{(1)}, \boldsymbol{X}^{(1)}\}$ and $\mathcal{G}^{(2)} = \{\mathcal{N}^{(2)}, \mathcal{E}^{(2)}, \boldsymbol{X}^{(2)}\}$, we generate two representation matrices $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{Z}^{(2)}$ via Eq. (3), and use $\boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)} \in \mathbb{R}^D$ to denote representation vectors of HCL n_i in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$, respectively.

In the two local graphs, we use the representations of the same HCL as a positive pair, and the representations of different HCLs as a negative pair. A non-linear projection function g_ξ is adopted to map the representations into another latent space for loss calculation, *i.e.,* $\boldsymbol{u}_i = g_\xi\left(\boldsymbol{z}_i^{(1)}\right)$ and $\boldsymbol{v}_i = g_\xi\left(\boldsymbol{z}_i^{(2)}\right)$. In the latent space, we employ the InfoNCE loss [12] to define our pairwise local-local contrastive loss as

$$L_{llc}(\boldsymbol{u}_i, \boldsymbol{v}_i) = -\log \frac{h(\boldsymbol{u}_i, \boldsymbol{v}_i)}{h(\boldsymbol{u}_i, \boldsymbol{v}_i) + \sum_{k \neq i} h(\boldsymbol{u}_i, \boldsymbol{u}_k) + \sum_{k \neq i} h(\boldsymbol{u}_i, \boldsymbol{v}_k)}. \quad (7)$$

Here h is a criterion function defined as $h(\boldsymbol{u}, \boldsymbol{v}) = \exp(\text{sim}(\boldsymbol{u}, \boldsymbol{v})/\tau)$, where $\text{sim}(\cdot, \cdot)$ is the cosine similarity and τ is an adjustable temperature parameter. We implement g_ξ using a two-layer MLP.

The final local-local contrastive loss of the two local graphs is defined as

$$L_{llc} = \frac{1}{2|\mathcal{N}^{(llc)}|} \sum_{n_i \in \mathcal{N}^{(llc)}} \left(L_{llc}(\boldsymbol{u}_i, \boldsymbol{v}_i) + L_{llc}(\boldsymbol{v}_i, \boldsymbol{u}_i)\right), \quad (8)$$

where $\mathcal{N}^{(llc)} = \mathcal{N}^{(1)} \cap \mathcal{N}^{(2)}$.

Contrastive Task #2: Local-Global Contrast. The *local-global contrast* task focuses on the representation consistency between local graphs and the global graph. Similar to the local-local contrast, we project representation $\boldsymbol{z}_i$ of HCL n_i in the global graph by g_ξ, *i.e.,* $\boldsymbol{w}_i = g_\xi(\boldsymbol{z}_i)$. The pairwise local-global contrastive loss between $\boldsymbol{u}_i$ and $\boldsymbol{w}_i$ is defined as

$$L_{lgc}(\boldsymbol{u}_i, \boldsymbol{w}_i) = -\log \frac{h(\boldsymbol{u}_i, \boldsymbol{w}_i)}{h(\boldsymbol{u}_i, \boldsymbol{w}_i) + \sum_{k \neq i} h(\boldsymbol{u}_i, \boldsymbol{u}_k) + \sum_{k \neq i} h(\boldsymbol{u}_i, \boldsymbol{w}_k)}, \quad (9)$$

the settings of which are the same as that for Eq. (7). The final local-global contrastive loss is

$$L_{lgc} = \frac{1}{2|\mathcal{N}^{(lgc)}|} \sum_{n_i \in \mathcal{N}^{(lgc)}} \left(L_{lgc}(\boldsymbol{u}_i, \boldsymbol{w}_i) + L_{lgc}(\boldsymbol{v}_i, \boldsymbol{w}_i)\right), \quad (10)$$

where $\mathcal{N}^{(lgc)} = \mathcal{N}^{(1)} \cup \mathcal{N}^{(2)}$.

5.2.3 *The Overall Objective.* We finally train the representation generator $f_\theta(\cdot)$ defined in Eq. (3) on the four tasks simultaneously. The overall loss function is defined as

$$L_{all} = L_{dp} + L_{cp} + L_{llc} + L_{lgc}, \quad (11)$$

where L_{dp} (Eq. (4)), L_{cp} (Eq. (6)), L_{llc} (Eq. (8)) and L_{lgc} (Eq. (10)) are the loss of degree prediction, context prediction, local-local contrast and local-global contrast, respectively.

To sum up, at each training epoch, we first draw two local HCL graphs. Then, we obtain HCL representations of the two local graphs and the global graph using the same GNN encoder $f_\theta(\cdot)$. Next, the representations are fed into the four tasks to calculate the overall loss in Eq. (11). Finally, the parameters are updated through backpropagation algorithm by minimizing the overall loss.

5.3 Risk Level Classification

The goal of risk classification is to predict HCL risk levels. In this component, we only keep and freeze the parameters of the GNN encoder $f_\theta(\cdot)$ to generate representations. Using the representation $\boldsymbol{z}_i$ for HCL n_i as an input, we employ a linear transformation with SoftMax activation to predict risk level of n_i as

$$\hat{\boldsymbol{y}}_i = \text{SoftMax}\left(\boldsymbol{W}_c \boldsymbol{z}_i + \boldsymbol{b}_c\right), \quad (12)$$

where $\hat{\boldsymbol{y}}_i$ is the predicted risk level, $\boldsymbol{W}_c \in \mathbb{R}^{K \times D}$ and $\boldsymbol{b}_c \in \mathbb{R}^K$ are learnable parameters. We adopt a cross-entropy loss as the objective function to train the learnable parameters, *i.e.,*

$$L_c = -\sum_{i, \boldsymbol{y}_i \text{ is known}} \sum_{k=1}^{K} \boldsymbol{y}_i^{(k)} \log\left(\hat{\boldsymbol{y}}_i^{(k)}\right), \quad (13)$$

where $\boldsymbol{y}_i \in \mathbb{R}^K$ is the real risk level of n_i. Both $\hat{\boldsymbol{y}}_i$ and $\boldsymbol{y}_i$ are in the one-hot encoding.

6 EXPERIMENTS

6.1 Datasets

We implemented and applied the CityShield system to the city of Nantong, China. Nantong is heavily dependent on the chemicals industry, which contributes 41.9% of the secondary industry GDP of Nantong in 2020[5]. Therefore, hazardous chemicals management is the top priority of Nantong's public safety. The trajectory data used in our experiments are collected from November 15 to December 31 in 2020 (denoted as NT20) and March 1 to April 15 in 2021 (denoted as NT21). Due to the GPS drifting issue, we filter the noise GPS points with speed threshold 100 km/h. For stay point detection, we set $d_{max} = 50$ meters and $t_{min} = 5$ minutes. After data preprocessing, we obtain 488,525 and 526,237 stay points for NT20 and NT21, respectively. The POI data include 235,273 POIs of 134 categories. The data set also contains an HCL registration list, which contains 2,287 supervised HCLs with their risk level labels. We use these HCLs for training and validation.

6.2 HCL Recognition

Through HCL-Rec algorithm, we obtain 4,696 and 5,261 HCLs for NT20 and NT21 with recognition radius $r = 20$ meters.

[4] Here, we name the HCL graph constructed by all trajectories as a global HCL graph

[5] http://www.stats.gov.cn/english/

We employ the recognition recall, *i.e.,* the fraction of the relevant HCLs that are successfully recognized, as a metric to evaluate HCL-Rec's performance. Because there are no ground truth for HCL boundaries, we manually draw the boundaries of some factories as ground truth. We consider an HCL was successfully recognized when the overlap between the HCL and its ground truth reaches 50% of the area of the HCL itself. Each ground truth can only be considered to be hit by the recognized HCL with the largest overlap. We compare the recall of our proposed HCL-Rec and DBSCAN. HCL-Rec achieves a recall of 0.944 and 0.916 for NT20 and NT21, while DBSCAN 0.906 and 0.891. DBSCAN performs worse than HCL-Rec because it is prone to merge adjacent HCLs into one, while HCL-Rec can accurately identify each of them. This ensures our CityShield can effectively discover HCLs for hazardous chemicals management.

Fig. 6 visualizes the HCLs recognized from the NT21 dataset. As depicted in Fig. 6(a), we present the spatial distribution of HCLs on the city map of Nantong. We can see that the HCL spatial distribution is skewed, *i.e.,* many HCLs are in small regions. To see more clearly, we zoom in an area in Fig. 6(b). The HCL distribution is dense in this area because there exists a chemical industrial park. From the recognition results shown in Fig. 6(c), the proposed HCL-Rec can accurately identify two neighbor HCLs, while DBSCAN merges them into one.

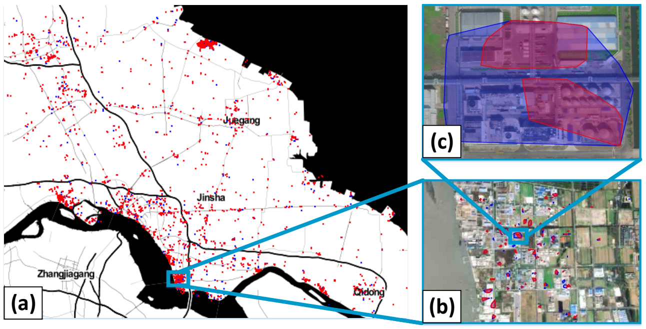


Figure 6: Spatial distribution of HCLs from NT21 dataset. Red and blue HCLs are the recognition results of our HCL-Rec and DBSCAN, respectively.

6.3 HCL Classification

This subsection demonstrates the HCL classification function of the CityShield system.

6.3.1 Settings. There are 419 and 443 labeled HCLs in NT20 and NT21 datasets, respectively. For the labeled HCLs, we randomly split them, where 70%, 10% and the rest 20% are selected for the training, validation and test set, respectively. For HCL graph construction, we use all trajectories to generate the global one and use each day's trajectories to generate local ones. The statistics are shown in Tab. 1. Note we report mean values for local HCL graphs.

Table 1: Statistics of HCL graph.

Dataset	NT20		NT21	
Metric	#Nodes	#Edges	#Nodes	#Edges
Global graph	4696	7048	5261	8115
Local graph (avg.)	474	532	565	648

Baselines & Evaluation Metrics. We consider the baselines belonging to the following three categories: *i*) Graph node embedding methods including DeepWalk [13] and node2vec [2]. *ii*) Graph deep learning based self-supervised methods including Graph Auto-Encoder (GAE, VGAE) [8], Deep Graph Infomax (DGI) [19], Bootstrapped Graph Latents (BGRL) [17], and GCA [29]. *iii*) Graph Convolutional Networks (GCN) [9] and Graph Attention Networks (GAT) [18] as the representatives of supervised learning counterparts. Since the HCL label distribution on different categories is imbalanced, we employ four unbiased metrics to evaluate our model, including mean per-class accuracy (MPA), matthews correlation coefficient (MCC), macro-averaged F1-score (F1) and area under the receiver operating characteristic curve (AUROC) [3].

We train our model for 100 epochs with early stopping strategy, and use Adam optimizer with weight decay. All hyperparameters are selected based on the performance on validation set.

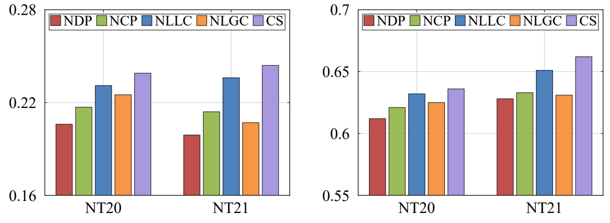
6.3.2 Performance on HCL Classification. The performance comparison of all methods on HCL classification are summarized in Tab. 2. Overall, CityShield outperforms other baselines, even the supervised counterpart GCN, with a large margin on the both datasets (average 7.1% improvement over the best baseline). Notably, our CityShield stably delivers the best results across different evaluation metrics, which verifies the superiority of the proposed method.

We make other observations as follows. *i*) Graph node embedding methods, *e.g.,* DeepWalk and node2vec, perform worst among all baselines because they assume that nearby nodes in graph are close in the representation space. However, in the HCL graph, nearby nodes usually belongs to different categories, violating the assumption of the these methods. *ii*) Graph deep learning-based supervised learning methods, such as GCN and GAT, require rich labels to training model. Since the label ratio of HCL is quite low (9.3% and 8.4% for NT20 and NT21) and imbalanced in our application, these methods are prone to overfitting, leading to unsatisfactory results. *iii*) Self-supervised methods take full advantage of abundant unlabeled data and usually perform better. In particular, methods based on contrastive tasks, *e.g.,* BGRL, DGI and GCA, which aim to extract additional supervision information from data similarities for improving the learned representations, obtain promising classification results. *iv*) In our CityShield, two different type of tasks, *i.e.,* predictive and contrastive, are adopted to enrich the supervision signals from graph structure and data similarities, which can help generate discriminative representations for the HCL classification task. Furthermore, in contrastive tasks, we introduce augmented data with real-world physical meanings (*i.e.,* local HCL graphs) rather than using random data augmentations. This eliminates unnecessary noises and then results in more stable performance.

6.3.3 Ablation Experiments. In the HCL representation learning component, we propose four self-supervised tasks in two types, including two predictive tasks, *i.e.,* degree prediction and context prediction, and two contrastive tasks, *i.e.,* local-local contrast and local-global contrast. Here, we determine how each component

Table 2: Summary of performance on HCL classification in terms of four evaluation metrics with mean and standard deviation. "↑" indicates "larger is better". The best performance is highlighted in bold, while the second best performance is underlined.

Dataset	NT20				NT21			
Metric	MPA ↑	MCC ↑	F1 ↑	AUROC ↑	MPA ↑	MCC ↑	F1 ↑	AUROC ↑
DeepWalk	0.181 ± 0.013	0.132 ± 0.031	0.175 ± 0.015	0.548 ± 0.022	0.160 ± 0.015	0.124 ± 0.021	0.138 ± 0.021	0.545 ± 0.020
node2vec	0.178 ± 0.011	0.125 ± 0.019	0.173 ± 0.012	0.542 ± 0.020	0.151 ± 0.012	0.110 ± 0.043	0.135 ± 0.013	0.534 ± 0.022
GCN	0.212 ± 0.018	0.211 ± 0.018	0.205 ± 0.016	0.581 ± 0.020	0.192 ± 0.015	0.165 ± 0.016	0.183 ± 0.015	0.579 ± 0.012
GAT	0.195 ± 0.012	0.216 ± 0.037	0.180 ± 0.014	0.591 ± 0.028	0.195 ± 0.011	0.199 ± 0.040	0.179 ± 0.010	0.580 ± 0.020
GAE	0.203 ± 0.013	0.195 ± 0.041	0.196 ± 0.014	0.567 ± 0.030	0.177 ± 0.019	0.211 ± 0.059	0.177 ± 0.020	0.584 ± 0.023
VGAE	0.171 ± 0.025	0.172 ± 0.071	0.159 ± 0.023	0.564 ± 0.023	0.155 ± 0.014	0.132 ± 0.051	0.143 ± 0.015	0.587 ± 0.023
BGRL	0.211 ± 0.008	0.219 ± 0.018	0.208 ± 0.009	0.620 ± 0.027	0.199 ± 0.014	0.238 ± 0.040	0.185 ± 0.015	0.615 ± 0.015
DGI	0.214 ± 0.024	0.220 ± 0.030	0.210 ± 0.022	0.621 ± 0.019	0.178 ± 0.021	0.186 ± 0.046	0.178 ± 0.025	0.602 ± 0.019
GCA	0.224 ± 0.018	0.214 ± 0.016	0.221 ± 0.016	0.614 ± 0.014	0.226 ± 0.018	0.192 ± 0.026	0.234 ± 0.027	0.651 ± 0.013
CityShield	**0.255 ± 0.011**	**0.237 ± 0.008**	**0.239 ± 0.012**	**0.636 ± 0.014**	**0.245 ± 0.013**	**0.263 ± 0.017**	**0.244 ± 0.01**	**0.662 ± 0.012**



(a) F1 (b) AUROC

Figure 7: Ablation study on four tasks. CS denotes CityShield.

contributes to the final performance. We compare four variants of the proposed CityShield: *i*) NDP without the degree prediction task, *ii*) NCP without the context prediction task, *iii*) NLLC without the local-local contrastive task, and *iv*) NLGC without the local-global contrastive task. The performance is reported in Fig. 7.

From Fig. 7, we observe that: *i*) On both datasets, NDP performs worst among all variants indicating the HCL degree being the most discriminative feature for different HCL categories. *ii*) NCP performs the second worst, which verifies the effectiveness of HCL neighbor information. *iii*) NLLC delivers better performance than NLGC because the global HCL graph contains more rich and complete HCL relations, so it can generate better HCL representations.

6.3.4 Case Study. We further give a case study to verify the effectiveness of CityShield in real-world applications. In this case, we select two unlabeled HCLs with the high risk levels from the NT21 dataset, *i.e.,* two unknown high risk HCLs for the authorities, which the HCL *a* with a predicted risk level of 7 (*i.e.,* predicted as production category) and the HCL *b* with a predicted risk level of 6 (*i.e.,* predicted as storage category), as depicted in Fig. 8. We reported the two HCLs to local hazardous chemicals management department. Through on-site investigations, the authorities found that HCL *a* was a factory that had resumed work illegally. This factory was shut down before due to illegal production. HCL *b* was an illegal storage warehouse hidden in a farmer's home. The investigation results match our prediction well, demonstrating the effectiveness of our CityShield in real-world scenarios.

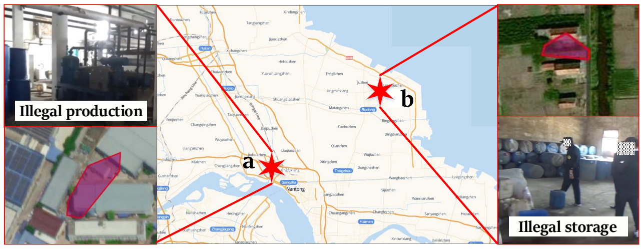


Figure 8: Case study of two unlabeled high-risk HCLs *a* and *b*, which are unknown to the authorities before. CityShield predicts *a* to be risk level 7 (the production category) and *b* to be risk level 6 (the storage category). On-site investigations validate our model's predictions.

7 SYSTEM DEPLOYMENT

CityShield has been deployed in the Modernized City Governance Platform of Nantong[6] since September 2021. The entire process of CityShield (described in Sec. 2.2) takes about two hours to perform, so the model is deployed offline in a daily-run mode. For each run, CityShield takes the latest 45 days trajectories. Until January 2022, our CityShield has analyzed 327.72 million GPS points and recognized 7,294 different HCLs. Among these HCLs, only 829 are known and under the supervision of the authorities, and the remaining 6,465 are unknown before. The authorities have inspected 289 of them due to limited resources and discovered 173 hidden dangers of hazardous chemicals.

8 RELATED WORK

Hazardous Chemicals Transportation. Transportation of hazardous chemicals has aroused widespread concern in hazardous chemicals management and intelligent transportation systems (ITS) areas. In hazardous chemicals management, studies mainly focus on transportation risk definition [5], risk factor identification [27] and accident analysis [24]. ITS researchers focus on transportation route planning [11] and traffic system designing [16]. Most of these works study the risk during the transportation process, which can

[6] https://www.zghy.org.cn/item/377846920866922496

be very random because of many uncertainties. Therefore, identifying high-risk locations related to hazardous chemicals becomes increasingly important in real-world applications. To identify these locations, traditional methods mainly rely on manual processes, such as random inspection, which is labor-intensive and inefficient. To remedy the deficiency, [20] and [21] propose to partition a city into grids and recognize risk locations through data-driven methods. In comparison, we propose to recognize locations with polygon boundaries, which contains more semantic information than grids. The boundary is also valuable for authorities to quickly determine a regulation area.

Graph Representation Learning. HCL graph representation learning is a key component of CityShield. Traditional methods mainly rely on random walk-based objectives, such as DeepWalk [13] and node2vec [2]. The basic assumption is that nearby nodes in the input graph should be "close" in the representation space. These methods over-emphasize proximity information [14], and the performance is highly dependent on hyperparameter choice [2, 13]. Recent work on graph neural networks has demonstrated impressive results, such as GCN [9] and GAT [18]. However, these methods require labeled dataset that may not be accessible in real-world applications. The other line aims to utilize abundant unlabeled data, called self-supervised graph representation learning. Methods of this line can be divided into predictive and contrastive. Predictive methods aims to extract supervision signals from graph topology [4], node attributes [7], and specific domain knowledge [15]. Contrastive methods aims to learn discriminative representations by contrasting positive and negative samples from stochastic graph augmentation [22], such as DGI [19] and GCA [29].These general approach did not fully consider the unique characteristics of HCL graphs, so can not be directly applied in HCL classification.

Urban Computing. Our work also falls into the research category of urban computing [26]. Urban computing aims to address the issues caused by the rapid urbanization, *e.g.*, public safety maintenance [30], urban anomaly detection [23], and traffic flow prediction [6]. This work focuses on maintaining the public safety by discovering high-risk locations caused by hazardous chemicals.

9 CONCLUSION

In this paper, we proposed a hazardous chemicals management system named as CityShield to mine HCLs from trajectories of hazardous chemicals transportation vehicles. To achieve precise HCL recognition and classification, we designed three components in the system. They are a *Data Pre-processing* component for trajectory denoising and transportation vehicle stay point detection, an *HCL Recognition* component using the proposed HCL-Rec algorithm incorporating vehicle ID information for accurate HCL recognition, and an *HCL Classification* component using pre-training-based representation learning to solve the label scarcity problem in HCL risk level and category classification. As a practically deployed system, the effectiveness of the CityShield was evaluated on offline real-world large-scale datasets and online hazardous chemicals management in the Nantong city. We believe the methods proposed in CityShield, such as elaborate pre-training tasks of location representation learning, are valuable other spatio-temporal data mining system designing.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2019YFB2101804), the National Natural Science Foundation of China (82161148011, 72171013, 62172034, 72031001) and the Beijing Nova Program (Z201100006820053). The authors would like to thank the following people for their helpful discussion: Hepeng Gao, Yanping Sun and Dayan Zhao.

REFERENCES

[1] M. Ester, H. Kriegel, J. Sander, X. Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *SIGKDD*. 226–231.

[2] A. Grover and J. Leskovec. 2016. node2vec: Scalable feature learning for networks. In *SIGKDD*. 855–864.

[3] D. Hand and R. Till. 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine learning* 45, 2 (2001), 171–186.

[4] Z. Hu, Y. Dong, K. Wang, K. Chang, and Y. Sun. 2020. Gpt-gnn: Generative pre-training of graph neural networks. In *SIGKDD*. 1857–1867.

[5] X. Huang, X. Wang, J. Pei, M. Xu, X. Huang, and Y. Luo. 2018. Risk assessment of the areas along the highway due to hazardous material transportation accidents. *Natural hazards* 93, 3 (2018), 1181–1202.

[6] J. Ji, J. Wang, Z. Jiang, et al. 2022. STDEN: Towards Physics-guided Neural Networks for Traffic Flow Prediction. In *AAAI*.

[7] W. Jin, T. Derr, Y. Wang, Y. Ma, Z. Liu, and J. Tang. 2021. Node similarity preserving graph convolutional networks. In *WWW*. 148–156.

[8] T. Kipf and M. Welling. 2016. Variational Graph Auto-Encoders. *stat* 1050 (2016), 21.

[9] T. Kipf and M. Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[10] Q. Li, Y. Zheng, X. Xie, et al. 2008. Mining user similarity based on location history. In *SIGSPATIAL*.

[11] M. Noureddine and M. Ristic. 2019. Route planning for hazardous materials transportation: Multicriteria decision making approach. *Decision making: applications in management and engineering* 2, 1 (2019), 66–85.

[12] A. Oord, Y. Li, and O. Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR* abs/1807.03748 (2018).

[13] B. Perozzi, R. Al-Rfou, and S. Skiena. 2014. Deepwalk: Online learning of social representations. In *SIGKDD*. 701–710.

[14] L. Ribeiro, P. Saverese, and D. Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *SIGKDD*. 385–394.

[15] Y. Rong, Y. Bian, T. Xu, et al. 2020. Self-supervised graph transformer on large-scale molecular data. *NeurIPS* (2020), 12559–12571.

[16] F. Santarremigia, G. Molero, S. Poveda-Reyes, and J. Aguilar-Herrando. 2018. Railway safety by designing the layout of inland terminals with dangerous goods connected with the rail transport system. *Safety Science* 110 (2018), 206–216.

[17] S. Thakoor, C. Tallec, M. Azar, R. Munos, P. Veličković, and M. Valko. 2021. Bootstrapped Representation Learning on Graphs. In *ICLR*.

[18] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. 2018. Graph Attention Networks. *ICLR*.

[19] P. Veličković, W. Fedus, W. Hamilton, P. Liò, Y. Bengio, and D. Hjelm. 2019. Deep Graph Infomax. In *ICLR*.

[20] J. Wang, C. Chen, J. Wu, and Z. Xiong. 2017. No longer sleeping with a bomb: a duet system for protecting urban safety from dangerous goods. In *SIGKDD*.

[21] J. Wang, X. Lin, Y. Zuo, et al. 2021. DGeye: Probabilistic Risk Perception and Prediction for Urban Dangerous Goods Management. *ACM TOIS* 39, 3 (2021), 1–30.

[22] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen. 2020. Graph contrastive learning with augmentations. *NeurIPS*, 5812–5823.

[23] H. Zhang, Y. Zheng, and Y. Yu. 2018. Detecting urban anomalies using multiple spatio-temporal data sources. *IMWUT* 2, 1 (2018), 1–18.

[24] X. Zhang and X. Li. 2018. Analysis of hazardous chemicals transportation accidents and transportation management. *Chemical Engineering Transactions* 67 (2018), 745–750.

[25] Y. Zheng. 2015. Trajectory data mining: an overview. *ACM TIST* 6, 3 (2015), 1–41.

[26] Y. Zheng. 2019. *Urban computing*. MIT Press.

[27] K. Zhou, G. Huang, S. Wang, and K. Fang. 2020. Research on transportation safety of hazardous chemicals based on Fault Tree Analysis (FTA). In *ICITM*. 206–209.

[28] X. Zhu, Z. Ghahramani, and J. Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*. 912–919.

[29] Y. Zhu, Y. Xu, F. Yu, et al. 2021. Graph contrastive learning with adaptive augmentation. In *WWW*. 2069–2080.

[30] Z. Zhu, H. Ren, S. Ruan, et al. 2021. ICFinder: A Ubiquitous Approach to Detecting Illegal Hazardous Chemical Facilities with Truck Trajectories. In *SIGSPATIAL*.