

CrowdPatrol: A Mobile Crowdsensing Framework for Traffic Violation Hotspot Patrolling

Zhihan Jiang, Hang Zhu, Binbin Zhou, Chenhui Lu, Mingfei Sun, Xiaojuan Ma, *Member, IEEE*, Xiaoliang Fan, *Senior Member, IEEE*, Cheng Wang, *Member, IEEE*, Longbiao Chen*, *Member, IEEE*

Abstract—Traffic violations have become one of the major threats to urban transportation systems, undermining human safety and causing economic losses. To alleviate this problem, crowd-based patrol forces including traffic police and voluntary participants have been employed in many cities. To adaptively optimize patrol routes with limited manpower, it is essential to be aware of traffic violation hotspots. Traditionally, traffic violation hotspots are directly inferred from experience, and existing patrol routes are usually fixed. In this paper, we propose a mobile crowdsensing-based framework to dynamically infer traffic violation hotspots and adaptively schedule crowd patrol routes. Specifically, we first extract traffic violation-prone locations from heterogeneous crowd-sensed data and propose a spatiotemporal context-aware self-adaptive learning model (CSTA) to infer traffic violation hotspots. Then, we propose a tensor-based integer linear problem modeling method (TILP) to adaptively find optimal patrol routes under human labor constraints. Experiments on real-world data from two Chinese cities (Xiamen and Chengdu) show that our approach accurately infers traffic violation hotspots with *F1-scores* above 90% in both cities, and generates patrol routes with *relative coverage ratios* above 85%, significantly outperforming baseline methods.

Index Terms—Traffic violation, urban computing, patrol task scheduling, mobile crowdsensing.

1 INTRODUCTION

TRAFFIC violations, such as speeding, illegal turning, illegal parking, bring potential congestions, accidents and parking problems and have increasingly become a significant challenge in urban transportation systems [1]. For example, in 2018, New York City issued 54,469 traffic violation tickets and reported 44,508 traffic-related injuries across the city [2]. Approximately 1.35 million people die every year across the world as a result of road traffic accidents, costing most countries 3% of their gross domestic product [3]. To mitigate this issue, traffic police are dispatched to patrol around the city and station at some road intersections or segments with frequent traffic violations in rush hours (as shown in Fig. 1). However, due to the limited manpower, traffic police usually cannot cover the entire road network [4]. Besides, due to the lack of knowledge about traffic violation hotspots, the patrol routes are usually fixed and suboptimal, leading to a waste of human labor and deteriorates the effectiveness of patrol [5]. Therefore, it is essential to have a comprehensive understanding of dynamic traffic violation hotspots, so as to adaptively schedule the patrol routes of traffic police and incorporate voluntary participants when available.

Fortunately, the widely deployed crowdsensing systems generate heterogeneous data and provide us unprecedented

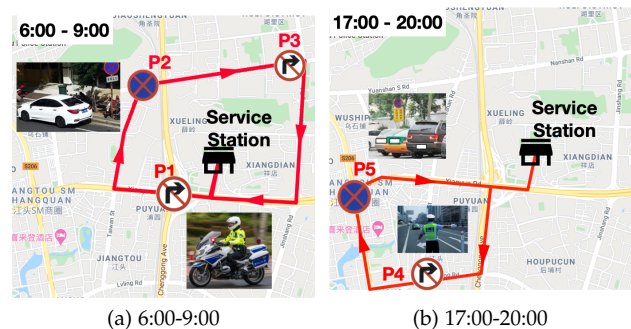


Fig. 1. An illustration for patrol participants conducting patrol for traffic violation hotspots.

opportunities for dynamic and efficient traffic violation hotspots patrolling in smart cities [6]. More specifically, city-wide traffic violation events can be extracted from these crowd-sensed data, and traffic violation hotspots can be inferred from traffic violation events. However, traditional methods usually infer traffic violation hotspots through statistics of historical violation events [7]–[9], failed to reflect the highly dynamic traffic violation hotspots accurately. Therefore, a *dynamic* and *comprehensive* method is in great demand for traffic violation hotspot inference.

With the dynamic traffic violation hotspots distribution, we need to design patrol routes adaptively. However, due to the limited manpower, traffic police usually cannot cover the entire road network, and their patrol routes are usually fixed. In recent years, many researchers have proposed to leverage the power of crowd to collaboratively perform large-scale complex sensing jobs [10], such as public information reposting [11], urban WiFi characterization [12], and road surface assessment [13]. Therefore, we propose to

- Z. Jiang, H. Zhu, C. Lu, X. Fan, C. Wang, L. Chen (the corresponding author) are with Fujian Key Laboratory of Sensing and Computing for Smart Cities (SCSC), School of Informatics, Xiamen University, China (e-mail: zhjiang@connect.hku.hk, longbiaochen@xmu.edu.cn).
- B. Zhou is with the Department of Computer Science and Computing, Zhejiang University City College (e-mail: bbzhou@zucc.edu.cn).
- X. Ma is with Hong Kong University of Science and Technology (e-mail: mxj@cse.ust.hk).
- M. Sun is with the University of Oxford (e-mail: mingfei.sun@cs.ox.ac.uk).

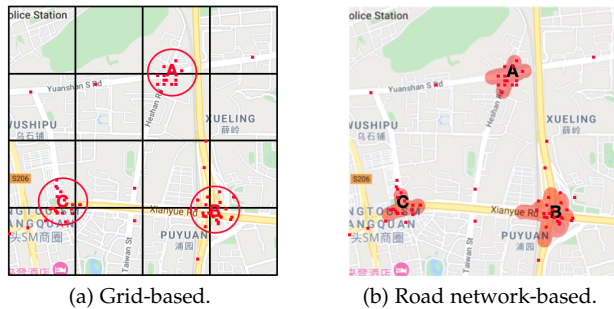


Fig. 2. An illustration of the advantages in road network-based traffic violation-prone location extraction. In (a), the road intersections A, B, and C are divided into different grids. In (b), A, B, and C are extracted as traffic violation-prone locations based on road network.

incorporate crowd volunteers for patrolling, and adaptively schedule patrol tasks to traffic police.

In this paper, we propose a mobile crowdsensing framework for traffic violation hotspot patrolling. We first extract traffic violation-prone locations and infer dynamic traffic violation hotspots from heterogeneous crowd-sensed data. Then, we incorporate city-wide mobile crowd users to estimate voluntary participants and adaptively schedule patrol tasks for traffic police. To implement this mobile crowdsensing framework, the following issues need to be addressed.

First, *it is non-trivial to extract traffic violation-prone locations from heterogeneous crowd-sensed data*. Since the heterogeneous crowd-sensed data, such as vehicle trajectories and road environments, are noisy and complex, it is difficult to extract traffic violation events. Moreover, traffic violation events are dynamically distributed across the road network. Traditional methods for traffic violation-prone location extraction, such as gridding the whole city and selecting grids with more frequent traffic violations [14], are inefficient, since many areas in cities are not roads. Besides, simply gridding the city may lead to the result that one road intersection or segment is divided into different grids (Fig. 2a), and it is also difficult to determine the size of grids. To address this issue, we propose to extract traffic violation-prone locations with dynamic boundaries (Fig. 2b) from crowd-sensed traffic violations based on the road network.

Second, *it is hard to infer traffic violation hotspots comprehensively*. Many researchers have been working on inferring traffic violation hotspots through analyzing drivers' behaviors [15], [16]. However, these methods focus on some key locations or drivers' personalities and are difficult to be transferred to the whole road network. Some researchers incorporate deep learning in traffic forecasting [17], [18], which usually requires a large amount of data and computing resources. Therefore, we propose to involve cross-domain road environment data to model the spatiotemporal contexts of traffic violation-prone locations, and combine tri-training and active learning (CSTA) to leverage the crowd intelligence for traffic violation hotspot inference.

Third, *it is difficult to optimize patrol routes effectively*. The effectiveness of patrol requires patrol routes to have high coverage ratios of traffic violation hotspots, which inevitably increases the workload of patrol participants. The designed patrol routes should keep a balance between effectiveness and workload. Also, the sub-route length should be limited to guarantee timely transfer between hotspots, and the pa-

trol routes should be vertex-disjoint so that different patrol participants would not station at the same location in the same time, avoiding wasting human resources. Therefore, we propose a tensor-based integer linear problem modeling method (TILP) incorporating the potential voluntary participants to find optimal vertex-disjoint patrol routes with a maximal coverage ratio of traffic violation hotspots under resource constraints.

To summarize, based on the above discussion, the contributions of this paper include:

- To the best of our knowledge, this is the first mobile crowdsensing framework to *dynamically* infer traffic violation hotspots and *adaptively* schedule patrol tasks. Such a *comprehensive* framework can help better understand spatiotemporal characteristics of traffic violation hotspots and schedule patrol tasks with higher effectiveness and fewer resources.
- We propose a three-phase framework to infer traffic violation hotspots in cities and schedule patrol tasks. Our method can not only *accurately* infer traffic violation hotspots by a context-aware self-adaptive learning model (CSTA) based on crowd-sensed traffic violations and contexts, but also *adaptively* schedule patrol tasks leveraging a tensor-based integer linear problem modeling method (TILP) incorporating mobile users.
- We evaluate our framework with *real-world* heterogeneous datasets from Xiamen and Chengdu. Results show that our framework can accurately infer traffic violation hotspots based on spatiotemporal contextual features with *F1-score* above 90% in both Chengdu and Xiamen, and achieves the best *relative coverage ratio* of 86.52% in Xiamen with voluntary participants, outperforming the baseline methods. We also deploy a real-world Traffic Violation Hotspot Patrolling Information Service System, as shown in Fig. 9

2 RELATED WORK

In this section, we first introduce the mobile crowdsensing, and then present an overview of the traffic violation inference and patrol task scheduling.

2.1 Mobile Crowdsensing

The mobile crowdsensing paradigm is one of the frontiers of mobile computing and has been widely studied in the literature. It takes advantages of the pervasive mobile devices and the power of crowd to facilitate large-scale sensing tasks that are costly or time consuming [19], [20]. Specifically, heterogeneous urban data, such as traffic violation events and weather conditions, can be collected through mobile crowdsensing systems.

It has been applied to many different areas and achieved good performance. For example, Chen et al. [21] proposed to plan city-wide package delivery paths leveraging crowd-sensed taxi trajectories. Guo et al. [11] presented a mobile crowdsensing system for cross-space public information reposting, tagging, and sharing. Wang et al. [22] proposed

a deep learning enabled industrial sensing and prediction scheme based on Sparse Mobile Crowdsensing.

In this paper, we propose a context-aware self-adaptive learning model combining tri-training and active learning based on mobile crowdsensing to infer traffic violation hotspots.

2.2 Traffic Violation Inference

Traffic violations refer to drivers' behaviors that violate traffic laws and have been a significant challenge in urban transportation systems [1]. There are plenty of studies on traffic violation inference [23], [24]. Artificial neural networks and deep learning approaches also have been used in traffic violation detection [25], [26]. Also, crowdsensing platforms for traffic violation reporting have been developed in many cities, such as Shanghai and Xiamen in China. With various of traffic violation inference technologies, the traffic violation data can be collected easier. However, few studies have been conducted on traffic violation hotspot modeling.

The context awareness computing is a mobile computing paradigm where the contextual information (such as locations and time) is gathered to improve the performance of systems or models [27]. Contexts have been leveraged in numerous studies, such as road obstacle classification [28] and search service [29], showing its potentiality and importance. In traffic violation hotspot inference, it is hard to infer traffic violation hotspots comprehensively merely based on violation data. Therefore, we involve cross-domain road environment data to model the spatiotemporal contexts of traffic violation-prone locations, and thus learn a classification model to infer traffic violation hotspots accordingly.

Furthermore, we combine tri-training [30] and active learning diagrams to address the challenge of insufficient labels and make the model adapt to dynamic environments. Tri-training algorithm employs three classifiers to provide pseudo-labels for unlabeled data to refine the classifiers and finally determine the labels of data with a voting mechanism. It has been used in authorship attribution [31] and unsupervised domain adaption [32], etc. Active learning improves the performance of model by dynamically selecting a set of ambiguous results and asking experts to label these instances, and then it feeds them back to retrain the model iteratively [33]. It has been used in recommender systems [34] and task allocation in crowdsensing [35], etc.

In this paper, we first extract traffic violation events from heterogeneous crowd-sensed data and cluster them into traffic violation-prone locations. Then, we infer traffic violation hotspots by combing tri-training and active learning based on the spatiotemporal contextual features.

2.3 Patrol Task Scheduling

The simplest scenario of patrol task scheduling can be regarded as a conventional routing problem whose objective is usually to select a path with minimal travel cost in a network or across multiple networks. Many algorithms, such as the Dijkstra algorithm and genetic algorithm, have been proposed to solve these problems. Chevaleyre et al. [36] regards this single-agent patrolling as the traveling salesman problem (TSP). In real-world application, patrol routes to be scheduled simultaneously are usually multiple. Hoffman

and Pavley introduced the k shortest paths problem for the first time [37]. Eilam et al. [38] introduces the disjoint shortest paths problem, and Tragoudas et al. [39] shows the problem of computing a pair of disjoint paths between a pair of nodes in an undirected graph is NP-complete.

Furthermore, many works have been proposed for various application scenarios in patrol task scheduling. However, many works [40]–[42] focus on the response performance to emergencies [43], such as traffic accidents, while the patrol routes designed for traffic violation hotspots give priority to high coverages and low costs. Leigh et al. [44] developed an algorithm to direct patrol routes by targeting high crime areas whilst maximizing demand coverage. However, this work focuses on the officers' allocation from police stations to crime hotspots and does not take the officers' transfer between hotspots into account. Li et al. [45] proposed an approach to obtain a near-optimal set of patrol routes based on the transition matrix updated by the cross entropy approach.

In this work, we propose a tensor-based integer linear programming modeling method to find optimal patrol routes with a maximal coverage ratio of traffic violation hotspots under resource constraints incorporating the power of voluntary participants.

3 PRELIMINARIES AND FRAMEWORK

Definition 1. Traffic Violation Events: the crowd-sensed traffic violation events can be described by a set of traffic violations denoted by 4-tuples, i.e., $P = \{p | p = (t, lat, lng, type)\}$, where $t, lat, lng, type$ are the time stamp, latitude, longitude, and type of traffic violations.

Definition 2. Traffic Violation-Prone Location: traffic violation-prone locations refer to locations with relatively more traffic violations compared to other locations in the road network. These locations might sometimes become traffic violation hotspots.

Definition 3. Traffic Violation Level: traffic violation levels refer to the intensity of traffic violations in traffic violation-prone locations during specific time periods. Higher traffic violation levels mean more frequent traffic violations. The traffic violation levels of a traffic violation-prone location varies with time. In this paper, traffic violation levels are evaluated hourly.

Definition 4. Traffic Violation Hotspot: traffic violation hotspots refer to traffic violation-prone locations during specific time slots when they have high traffic violation levels. In this paper, we divide traffic violation levels into three categories, detailed in the Evaluation Section.

Definition 5. Patrol Route and Sub-Patrol Route: a patrol route starts from an origin to a sequence of locations and ends at a destination. Patrol routes are **vertex-disjoint**¹ meaning that different patrol participants would not station at the same location during the same time. A sub-patrol route of a patrol route is the route from one patrol node to another patrol node in this route.

1. More descriptions about the vertex-disjoint constraint can be found in Appendix 3

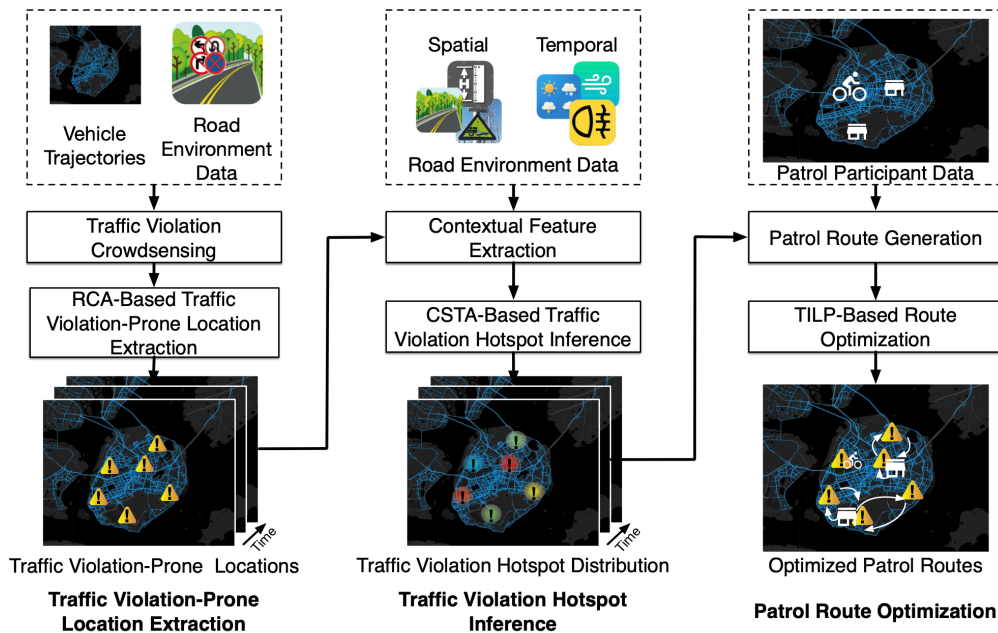


Fig. 3. Framework overview.

Definition 6. Patrol Participants: patrol participants are people conducting patrol tasks. We define two groups of patrol participants. The first group includes people who are obligated to conduct patrol tasks such as traffic police, and the second group are voluntary participants such as mobile crowd users.

We propose a three-phase mobile crowdsensing framework to dynamically infer traffic violation hotspots and adaptively schedule patrol tasks. As shown in Fig. 3, we first extract traffic violation events from heterogeneous crowdsensed data and cluster them into traffic violation-prone locations based on RCA. Then, we extract contextual features from road environments for traffic violation-prone locations and infer violation hotspots based on a CSTA model. Finally, we propose a TILP-based method to find the optimal patrol routes with a maximal coverage ratio of traffic violation hotspots under resource constraints for traffic police and incorporate voluntary participants when available.

4 TRAFFIC VIOLATION-PRONE LOCATION EXTRACTION

The development of crowdsensing systems enables the collection of large-scale heterogeneous data, such as vehicle trajectories, street view pictures, and road network data. These data provide us with great opportunities to extract traffic violation events and explore traffic violation-prone locations in cities. However, traffic violations are scattered across the whole road network. It is unrealistic and unreasonable to predict when and where a specific traffic violation event may happen and design patrol routes to prevent each single traffic violation incident. Instead, since traffic violations are highly dependent on the road network, they can be grouped according to their corresponding road segments or intersections, such as illegal turnings happening in the same intersection and illegal parking behaviors on the same road segment. Besides, although traffic violations are

scattered across the whole road network, they are not scattered uniformly. Some locations have relative more traffic violations than others, and due to limited patrol resources (e.g., traffic police and voluntary participants), only those traffic violation-prone locations with more traffic violations than other locations should be considered when scheduling patrol tasks for patrol participants. In this section, we first extract traffic violation events from heterogeneous crowdsensed data, and then introduce a road network-based clustering algorithm (RCA) to cluster scattered traffic violations into traffic violation-prone locations.

4.1 Traffic Violation Crowdsensing

We extract traffic violation events from large-scale crowdsensed vehicle trajectories and road environments leveraging the method proposed by [7]. Specifically, we first normalize the vehicle trajectories by mapping the trajectories onto the road network leveraging a map matching algorithm based on Hidden Markov Model [46], and then extract three types of driving behaviors, i.e., turning behaviors, parking behaviors, and speeds of vehicles from the normalized trajectories. Second, we model driver perspectives based on regression to restore spatiotemporal contexts of driving behaviors. After that, we train a YOLOv3-based traffic sign detector to detect traffic signs from corresponding street view images and extract road speed restrictions from map open platform. Finally, through matching the traffic restriction information with driving behaviors, we identify three types of traffic violations (i.e., illegal turning, illegal parking, and speeding). More details can be found in [7].

4.2 Road Network-Based Clustering Algorithm

The proposed algorithm is described in Algorithm 1. First, we obtain the road network data from OpenStreetMap², and divide the road network into two cate-

2. <http://www.openstreetmap.org/>

Algorithm 1: The road network-based clustering algorithm (RCA).

Input: R : The set of *intersections* and *segments*
 P : The set of traffic violations
 δ : The distance threshold
Output: S : The set of clusters

- 1 **for** $p \in P$ **do**
- 2 assign p to nearest $r \in R$
- 3 $S \leftarrow$ select $r \in R$ with p
- 4 $S_x \leftarrow$ sort S according to longitude
- 5 $S_y \leftarrow$ sort S according to latitude
- 6 $d_{min} \leftarrow \infty$, $d_{min} \leftarrow$ closest(S_x, S_y, d_{min})
- 7 **while** $d_{min} < \delta$ **do**
- 8 remove the two closest elements r_i, r_j from S
- 9 $r_{new}.lat \leftarrow (r_i.lat + r_j.lat)/2$
- 10 $r_{new}.lng \leftarrow (r_i.lng + r_j.lng)/2$
- 11 $r_{new}.tv \leftarrow r_i.tv + r_j.tv$
- 12 add r_{new} into S
- 13 $S_x \leftarrow$ sort S according to longitude
- 14 $S_y \leftarrow$ sort S according to latitude
- 15 $d_{min} \leftarrow \infty$, $d_{min} \leftarrow$ closest(S_x, S_y, d_{min})

gories $\{intersection, segment\}$. *Intersection* is the road intersection and *segment* is the road segment. R is the set of *intersections* and *segments*. P is the set of traffic violations. For each traffic violation $p \in P$, if it is an illegal turning behavior, we map it to the nearest *intersection*, and if it is illegal parking or illegal speeding, we assign it to the nearest *segment*. After that, we select the *intersections* or *segments* with traffic violations, denoted as a set of 3-tuples, $S = \{r | r = (lat, lng, tv)\}$, where *lat*, *lng*, *tv* are the average latitude, average longitude, and the set of traffic violations in *intersection* or *segment* r .

After getting S , we find the closest pair in S and if the distance of the closest pair (r_i, r_j) is less than a distance threshold δ , r_i and r_j are combined into one new element r_{new} . The latitude of r_{new} is the average latitude of r_i and r_j , and the longitude of r_{new} is the average longitude of r_i and r_j . The sets of traffic violations in r_i and r_j are also combined into the set of traffic violations in r_{new} . r_i and r_j are removed from S , and r_{new} is added into S . This process is repeated until the distance of the closest pair in S is no less than δ . The algorithm *closest* to find the closest two locations on the road network is detailed in Appendix 1.

After the clustering, the scattered traffic violations are clustered into traffic violation-prone locations. The reason for using the distance-based clustering is that the patrol participants can only cover a limited distance around them when stationing at a location. So traffic violations within this distance can be monitored by one patrol group, and those out of this distance should be monitored by other groups. Patrol tasks can thereby be assigned in units of traffic violation-prone locations.

Suppose there are n elements in S , the time complexity of the algorithm *closest* used in this paper is $\mathcal{O}(n \log n)$, and the time complexity of the Algorithm 1 is $\mathcal{O}(n^2 \log n)$ (detailed in Appendix 1). In particular, this algorithm is scalable. More specifically, if new traffic violations are added, we first assign them to the nearest r , if r belong to one cluster s in the current cluster set S , we add new traffic violations into s and update properties of s . If r does not

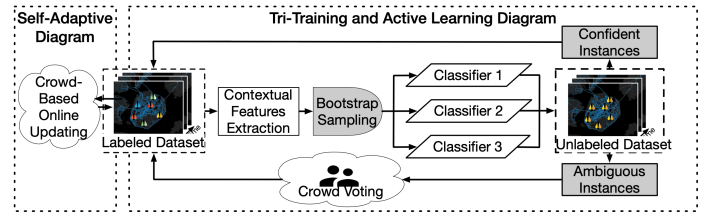


Fig. 4. The context-aware self-adaptive learning model (CSTA).

belong to any cluster in S , we can combine r with the current cluster set S , and input the new cluster set into the algorithm to update the cluster set. Besides, for each traffic violation-prone location, we can aggregate its traffic violations every time span³ to get its temporal pattern for traffic violations. In this paper, we aggregate its traffic violations and get its traffic levels hourly, detailed in the Evaluation Section.

5 TRAFFIC VIOLATION HOTSPOT INFERENCE

In order to schedule patrol tasks with a maximal coverage ratio of traffic violation hotspots timely, we need to infer dynamic traffic violation hotspots (i.e., traffic violation-prone locations with high traffic violation levels). Generally, such hotspots are more likely to be added to patrol routes. However, since the distribution of traffic violations is uneven, it is in-comprehensive to infer hotspots merely based on the traffic violation data. Therefore, we incorporate cross-domain road environment data to model spatiotemporal contexts of traffic violation-prone locations, and then infer traffic violation hotspots based on the contextual features.

In particular, we extract two categories of contextual features from road environment data. The first category is temporal contexts, such as date and meteorological conditions. For example, traffic violations may happen frequently in rush hours but rarely in other time periods, and more in rainy days than in sunny days. The second is spatial contexts, such as points of interest around and road categories. For example, a shopping mall nearby may cause frequent illegal parking on a road. After the extraction of contextual features, intuitively, we can train a classifier to infer traffic violation hotspots. However, it is difficult and time-consuming to obtain a labeled dataset (i.e., the traffic violation levels of traffic violation-prone locations) with size large enough for training a classifier, since there are many traffic violation-prone locations and each violation-prone locations have traffic violation levels vary across time.

To address the challenges of insufficient labels and adapt the model to dynamic environments, we propose a context-aware self-adaptive learning model combining tri-training and active learning (CSTA). We first use a bootstrap sampling method to generate three training datasets from the labeled dataset, and then input them into tri-training framework. We use a crowd-based active-learning mechanism to vote for labels for ambiguous instances generated in the tri-training process and gradually adapt the model to local environments. The design of CSTA model is illustrated in Fig. 4. We elaborate on contextual feature extraction and the CSTA model as follows.

3. The minimal time span for scheduling patrol tasks, determined by specific application scenarios.

5.1 Contextual Feature Extraction

5.1.1 Temporal Contextual Feature

The violation level of a traffic violation-prone location may vary significantly with different temporal contexts. For example, the traffic is usually heavy in rush hours, and thus there are usually more traffic violations, and in appalling weather, drivers may fail to see traffic signs clearly and violate traffic regulations. In particular, we extract the following two groups of key temporal contextual features that may relate to traffic violations from the corresponding road environment data.

Date and Time Features: Traffic violations in a location show periodic characteristics, and traffic violation hotspots with similar temporal patterns may have similar violation levels. Through incorporating the date and time features, hidden temporal dependencies among traffic violation hotspots can be captured. Therefore, we extract the *hour of a day, day of a week, day of a year, day of a month, month of a year, week of a year, and the quarter* as the date and time features. These features can be generated automatically.

Meteorological Features: The meteorological features can influence the visibility of traffic signs and roads as well as the mood of drivers related to traffic violations. Therefore, we extract the *temperature, dew point, humidity, wind speed, and pressure* as meteorological features. These features can be extracted from the meteorological data collected from the Weather Underground API⁴.

5.1.2 Spatial Contextual Feature

Based on the previous studies and surveys, the geographic environment of a location can have a significant influence of potential traffic violations [47]. For example, a location surrounded by many restaurants may see a lot of illegal parking behaviors, and a road segment with low speed limit may witness a lot of speeding. In particular, we extract the following two groups of key spatial contextual features that may relate to traffic violations from the corresponding road environment data.

POI Features: Point of interest (POI) is a specific location that someone may find useful or interesting, and different categories of POIs have different influences on the traffic violations nearby. We divide POIs into five categories, *business, accommodation, entertainment, infrastructure, tourism*. The *business* category includes the enterprises, business buildings and other business points; the *accommodation* category includes the restaurants, hotels, communities and other accommodation points; the *entertainment* category includes the shopping centers, cinemas, bars and other entertainment points; the *infrastructure* category includes the hospitals, banks, stations, government agencies and other infrastructure points; the *tourism* includes the tourist spots. We count the number of POIs for each category and build the POI features. The POI data are collected from BaiduMap⁵.

Location Features: For each traffic violation-prone location, it has specific *latitude, longitude, road category, and speed limit* that may influence its traffic violation patterns. For example, there are usually few traffic violations on broad highways, while on remote roads, traffic violations

4. <https://www.wunderground.com/weather/api/>
5. <https://map.baidu.com/>

Algorithm 2: The tri-training and active learning algorithm.

Input: D_L^0 : Labeled dataset, D_U^0 : Unlabeled dataset, ϵ : The label ability of crowd, *Learn*: Learning algorithm
Output: D_L : Labeled dataset, C_1, C_2, C_3 : The learned classifiers

- 1 **Initialize:** $D_U \leftarrow D_U^0, A \leftarrow \emptyset$
- 2 **for** $i \in \{1, 2, 3\}$ **do**
- 3 $D_L^i \leftarrow \text{BootstrapSampling}(D_L^0), C_i \leftarrow \text{Learn}(D_L^i)$
- 4 **while** $C_i, \forall i \in \{1, 2, 3\}$ has changed **do**
- 5 **for** $i \in \{1, 2, 3\}$ **do** $L_{ui} \leftarrow C_i(D_U)$
- 6 **for** $k \in D_U$ **do**
- 7 a) Select confident instances :
8 **if** $L_{u1}(k) = L_{u2}(k) = L_{u3}(k)$ **then**
9 add $L_{u1}(k)$ into D_L^1, D_L^2, D_L^3
- 10 b) Select relatively confident instances:
11 **if** $L_{ua}(k) = L_{ub} \neq L_{uc}$ **then**
12 $(a, b, c \in \{1, 2, 3\}, a \neq b \neq c)$
13 add $L_{ua}(k)$ into D_L^c
- 14 c) Select ambiguous instances:
15 **if** $L_{ua}(k) \neq L_{ub} \neq L_{uc}$ **and** $\text{size}(A) < \epsilon$ **then**
16 $(a, b, c \in \{1, 2, 3\}, a \neq b \neq c)$
17 add $\text{subsample}(k)$ into A
18 $L_A \leftarrow \text{Crowd_Voting}(A)$
19 add L_A into D_L^1, D_L^2, D_L^3
- 20 **for** $D_L^i (i \in \{1, 2, 3\})$ who has changed **do**
21 $C_i \leftarrow \text{Learn}(D_L^i)$
- 22 **for** $i \in \{1, 2, 3\}$ **do** $L_{ui} \leftarrow C_i(D_U^0)$
- 23 $D_L \leftarrow \text{voting}(L_{u1}, L_{u2}, L_{u3})$

may frequently happen. Therefore, we extract the *latitude, longitude, road category, and speed limit* as location features. The *road category* and *speed limit* data can be collected from Gaode Map Open Platform⁶.

5.2 Traffic Violation Hotspot Inference

5.2.1 Tri-Training and Active Learning Paradigm

Due to the label insufficiency, we combine tri-training and active learning in our learning model. We first use a bootstrap sampling method to generate three training sets and train three classifiers. In each round of tri-training, the instances that receive the same labels from three classifiers with high confidence are added to the training dataset, and the instances that receive the same labels from two classifier are labeled for the third classifier. The classifiers would be retrained if their corresponding training datasets have been updated. Besides, for instances that receive the different labels from all three classifiers, they are sampled randomly and added to an ambiguous dataset to be labeled by crowd voting, and the size of the ambiguous dataset is limited by the label ability of crowd. This process is iterated until none of the learned classifiers updated, and the final labels of the unlabeled data are determined by majority voting. The detailed process is described in Algorithm 2.

D_L^0 is the labeled dataset including the above-mentioned features of traffic violation-prone locations and labels (i.e., traffic violation levels). D_U^0 is the unlabeled dataset. ϵ is the label ability of crowd. First, a bootstrap sampling method is used to generate three training sets from D_L^0 , and based

6. <https://lbs.amap.com>

on these three training sets, we train three multi-class classifiers. Since the classifier is updated frequently and is capable of dealing with heterogeneous features, we adopt the XGBoost [48] in this paper. Then we use the classifiers to label the unlabeled dataset D_U . For each instance k in D_U , if it receive the same labels from three classifiers, it is regard as confident instance and is added to the training sets of the three classifiers. If it receive the same labels from two of three classifiers, it is regarded as relatively confident instance and its label from the two classifiers is added to the third one. If it receive different labels from all three classifiers, it is randomly added to an ambiguous dataset A . If the size of A does not exceed the label ability of crowd ϵ , the ambiguous instance would be labeled by crowd. Then for classifiers whose training set has been updated, the classifier would be retrained. This process is iterated until none of the learned classifiers updated. Finally, we use the three learned classifiers to label the unlabeled datasets with confidence. For each unlabeled instance, we aggregate its label results from all classifiers and assign the label with highest confidence to it.

5.2.2 Self-Adaptive Inference

The road environment may gradually change over time, thus our learning model works in an online manner, so it is adaptive to the current road environment. More specifically, given new dataset D_L^s collected by crowd (i.e., traffic police or mobile crowd users) in a period of time T , we add them into the current training set D_L and drop the data in the first T of the time span in D_L , and then update the model. Or given a new traffic violation-prone location l , we apply the classifier C_1 , C_2 and C_3 on l separately and determine its label by voting mechanism. If it gets the same label from no less than two classifiers, we add l into the training dataset D_L and retrain the classification model. If it receives totally different label from three classifiers, we ask crowd to vote for its final label and retrain the model. Since the online model is updated frequently, highly scalable, easy to train, and is capable of dealing with heterogeneous features, we adopt the XGBoost [48] as the ideal multi-class classifiers in this paper. This algorithm is also adaptive to other learning algorithms such as AdaBoost [49] and lightGBM [50].

6 PATROL ROUTE OPTIMIZATION

In this section, our objective is to adaptively schedule patrol tasks with a maximal coverage ratio of traffic violation hotspots to patrol participants (i.e., traffic police or voluntary participants) under resource constraints based on the traffic violation hotspot distribution. In patrol task scheduling, the compromise between the coverage ratio of traffic violation hotspots and route length as well as sub-route length is a great challenge to be addressed. The effective of patrol requires high coverage ratio of traffic violation hotspots of patrol routes, while the length of patrol routes and sub-patrol routes should be constrained so as to reduce travel costs and guarantee timely responses. Moreover, the number of routes is consistent with the number of patrol participants that work independently, which means that more than one routes are required to be allocated and there should not exist any overlapping in their routes except for

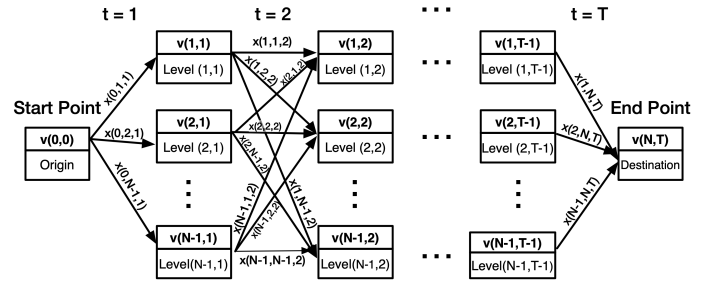


Fig. 5. The structure of the directed graph G modeled from the distribution of traffic violation hotspots.

the origin and the destination. Besides, the traffic violation hotspots are dynamic, which means each location can not be regarded as one fixed vertex. Also, when optimizing patrol routes, we incorporate the potential crowd power to further improve the coverage of traffic violation hotspots.

Therefore, we propose a tensor-based integer linear programming modeling method (TILP) to find optimal vertex-disjoint patrol routes with a maximal coverage ratio of traffic violation hotspots under route length, sub-route length, and vertex-disjoint constraints. First, based on the traffic violation hotspot distribution, we generate patrol routes based on a directed graph and model the optimal k patrol route problem whose objective is to find the optimal routes with a maximal coverage ratio of traffic violation hotspots under route length, sub-route length, and vertex-disjoint constraints. Then, we generate two groups of patrol participants including traffic police and voluntary participants. We estimate the potential voluntary participants for traffic violation hotspots based on the spatiotemporal patterns of bike returning events in the bike sharing system. Finally, we build corresponding adjacency tensor, gain tensor, cost tensor, crowd tensor to transfer the optimization problem into an integer linear programming problem and find the optimal solution.

6.1 Patrol Route Generation

Based on the distribution of traffic violation hotspots, we generate possible patrol routes with a directed graph $G = (V, X)$, as shown in Fig. 5, where $V = \{v(0,0), v(2,1), \dots, v(N-1,1), v(1,2), \dots, v(N-1,T-1), v(N,T)\}$, $v(n,t)$ ($n \in [1, N-1], t \in [1, T-1]$) denotes the node corresponding to the traffic violation hotspot n at time t , $v(0,0)$ and $v(N,T)$ denote the origin and destination of patrol routes, respectively. For each node $v(n,t)$, it has an attribute $Level(n,t)$ representing the violation level of traffic violation hotspot n at time t .

A route i from $v(0,0)$ to $v(N,T)$ can be denoted by a sequence of nodes, $v(0,0) \rightarrow v_{(q_1,1)}^i \rightarrow v_{(q_2,2)}^i \rightarrow \dots \rightarrow v_{(q_{T-1},T-1)}^i \rightarrow v(N,T)$, where $1 \leq q_1, q_2, \dots, q_{T-1} \leq N-1$, and for each node $v_{(n,t)}^i$, it has a violation level $Level_{(n,t)}^i$. The length of route i is denoted as l_i . Our objective is to find optimal k routes between $v(0,0)$ and $v(N,T)$ so that these routes have the maximal coverage ratio of traffic violation hotspots weighted by their violation levels under the constraints that their overall length is no longer than a threshold μ and the k routes are vertex-disjoint except for $v(0,0)$ and $v(N,T)$. Furthermore, to guarantee that a police officer can move

from one node to another timely, the distances between each pair of successive nodes are limited by a threshold λ .

With the above-mentioned definitions, we model the k patrol route optimization as follows.

Problem: (Optimal k Patrol Route Modeling)

$$\text{maximize } \sum_{i=1}^k \sum_{t=1}^{T-1} Level_{(q_i,t)}^i \quad (1)$$

subject to

$$v_{q_i,t}^i \neq v_{q_j,t}^j, \forall i, j \in [1, k], i \neq j, t \in [1, T-1] \quad (2)$$

$$\sum_{i=1}^k l_i \leq \mu \quad (3)$$

$$\forall i \in [1, k], t \in [0, T-1],$$

$$dist(v_{q_i,t}^i, v_{q_{i+1},t+1}^i) \leq \lambda \quad (4)$$

The objective 1 is to maximize the coverage ratio of traffic violation hotspots weighted by their violation levels. The requirement 2 represents that these k routes are vertex-disjoint except for the start and end nodes. The requirement 3 represents that the overall length of the k routes is no longer than a threshold μ . The requirement 4 represents that for each pair of successive nodes in patrol routes, the route length between them is no longer than a threshold λ .

6.2 Patrol Participant Generation

We divide the patrol participants into two groups. People in the first group are obligated to conduct patrol tasks, such as traffic police. They usually have fixed origins, destinations, and working hours, and their origins and destinations are in the same locations (e.g., traffic police stations), meaning that $v(0,0)$ is equal to $v(N,T)$.

The second group are voluntary participants willing to conduct patrol tasks. With the intuition that people who have just arrived at their destinations may willing to spare some time to look around, we incorporate mobile crowd users of bike sharing systems to collaboratively patrol with traffic police. Since there is no specific station in bike sharing systems, the bike returning events almost distribute across the whole road network. For a traffic violation hotspot, the more bike returning events happen nearby simultaneously, the more the potential crowdsensing participants exist for this hotspots, so the traffic police are more likely to off load this patrol task to voluntary participants.

More specifically, for each location, we extract the weekly spatiotemporal patterns of bike returning events within a distance threshold τ . For node $v_{(n,t)}$ in G , the expectation of bike returning events $Crowd_{(n,t)}$ is defined as the average number of bike returning events during the same time slot in the same day of a week in the weekly spatiotemporal patterns, and the probability of successfully recruiting a voluntary participant for node $v_{(n,t)}$ is defined as $p_{(n,t)}^{crowd} = 1 - (p_{rej})^{Crowd_{(n,t)}}$, where p_{rej} is the probability of a person rejecting the patrol task.

6.3 TILP-Based Patrol Route Optimization

Based on the optimal patrol route modeling above, we transfer this optimization problem to an integer linear programming problem through four tensors. We elaborate the

details of the three tensors as follows, and the illustration can be found at Appendix 2.

Adjacency Tensor: We use a $((N+1) \times (N+1) \times T)$ -tensor X to denote edges of graph G , where $x_{(i,j,t)} \in \{0,1\}$ denotes the edge between traffic violation-prone location i and location j at time t . If $x_{(v,w,t)} = 1$, it means that there is a transfer from location i to location j at time t , otherwise, there is no transfer.

Gain Tensor: We use a $((N+1) \times (N+1) \times T)$ -tensor L to denote the gain of moving from one location to another, where $l_{(i,j,t)} = Level_{(j,t)}$, meaning the traffic violation level of location j at time t .

Cost Tensor: We use a $((N+1) \times (N+1) \times T)$ -tensor D to denote the cost of moving from one location to another, where $d_{(i,j,t)} = dist(i,j)$, meaning the route distance from location i to j . The route distance is the shortest length of road segments from one location to another.

Crowd Tensor: We use a $((N+1) \times (N+1) \times T)$ -tensor B to denote the potential voluntary participants, where $b_{(i,j,t)} = p_{(j,t)}^{crowd}$. The value of $b_{(i,j,t)}$ increases with the rise of expected voluntary participant number of node $v_{(j,t)}$, and $b_{(i,j,t)} \in (0,1]$. If voluntary participants are out of consideration, all elements in B are set to 0.

Then our optimization problem can be rewritten as an integer programming problem with the objective of maximizing the gain under the travel cost and vertex-disjoint constraints.

Problem: (Integer Programming Problem)

$$\arg \max_X L \cdot X + \frac{\phi(B) \cdot \phi(L) \cdot (1 - \phi(X))}{(N+1)^2} \quad (5)$$

subject to

$$\sum_{j=1}^N x_{0,j,1} = \sum_{i=0}^N \sum_{j=0}^N x_{i,j,1} = \sum_{j=0}^N \sum_{t=1}^T x_{0,j,t} = k \quad (6)$$

$$\sum_{i=0}^N x_{i,N,T} = k \quad (7)$$

$$\sum_{i=0}^N \sum_{t=1}^{T-1} x_{i,0,t} = \sum_{i=0}^N \sum_{t=1}^{T-1} x_{i,N,t} = 0 \quad (8)$$

$$\forall t \in [1, T-1], p \in [1, N-1],$$

$$\sum_{i=0}^{N-1} x_{i,p,t} = \sum_{j=0}^{N-1} x_{p,j,t+1} \leq 1 \quad (9)$$

$$D \cdot X \leq \mu \quad (10)$$

$$\forall i \in [0, N], j \in [0, N], t \in [1, T],$$

$$x_{i,j,t} \in \{0,1\}, d_{i,j,t} x_{i,j,t} \leq \lambda \quad (11)$$

The objective of this problem 5 is to find the X that maximizes $L \cdot X + \frac{\phi(B) \cdot \phi(L) \cdot (1 - \phi(X))}{(N+1)^2}$, where $L \cdot X = \sum_{i=0}^N \sum_{j=0}^N \sum_{t=1}^T b_{i,j,t} l_{i,j,t} x_{i,j,t}$ representing the expected coverage by the first group of patrol participants. $\frac{\phi(B) \cdot \phi(L) \cdot (1 - \phi(X))}{(N+1)^2}$ representing the expected coverage by the second group, where $\phi(B)$ is a $(N+1) \times T$ -matrix, $\phi(B)_{j,t} = \sum_{i=0}^N b_{i,j,t}$. The definitions of $\phi(L)$ and $\phi(X)$ are similar to $\phi(B)$.

For requirement 6, $\sum_{j=1}^N x_{0,j,1}$ represents the number of routes starting from the origin when $t = 1$; $\sum_{i=0}^N \sum_{j=0}^N x_{i,j,1}$ represents the number of routes starting when $t = 1$;

TABLE 1
Summary of Traffic Violation Data and Road Environment Data

| Data type | Item | Chengdu | Xiamen |
|------------------------------|-----------------|--|---|
| Traffic Violations | Illegal parking | 14,790 | 8,184 |
| | Illegal turning | 504,872 | 627,279 |
| | Speeding | 1,198,941 | 109,737 |
| | POI | 834 | 12,228 |
| Environment Data | Road category | main road, minor road, non-navigational road, ordinary road urban expressway | main road, minor road, non-navigational road, ordinary road urban expressway, provincial highway, national highway, country highway |
| | Speed limit | 5 km/h, 30 km/h, 40 km/h, 50km/h, 60 km/h, 70 km/h, 80 km/h | |
| | Meteorological | temperature, dew point, humidity, wind speed, pressure (every hour) | |
| Geographic coverage (WGS-84) | | [30.655191,104.129591], [30.652828,104.042102] [30.727818,104.043333], [30.726490,104.129076] | [24.423250,118.064743], [24.423250,118.198504] [25.561485,118.064743], [24.561485,118.198504] |

$\sum_{j=0}^N \sum_{t=1}^T x_{0,j,t}$ represents the the number of routes starting from the origin. Therefore, this requirement represents that all k routes starts from $v_{(0,0)}$ when $t = 1$. For requirement 7, $\sum_{i=0}^N x_{i,N,T} = k$ represents that all k routes end at $v_{(N,T)}$. For requirement 8, $\sum_{i=0}^N \sum_{t=1}^{T-1} x_{i,0,t} = \sum_{i=0}^N \sum_{t=1}^{T-1} x_{i,N,t} = 0$ represents that no route would return to the origin location or end at destination before the end time. Requirements 6, 7, and 8 together guarantee the basic structure of the Graph G .

The requirement 9 represents that the number of groups that transfers to $v_{(p,t)}$ is equal to the number of that leaves this *node*, and this number is no more than one. This formula guarantees that no more than one group will travel to the same location at the same time except for the origin and destination, satisfying the vertex-disjoint constraint.

For requirements 10 and 11, since $x_{(i,j,t)}$ and $d_{(i,j,t)}$ denote the connectivity and route distance between *node* $v_{(i,t-1)}$ and $v_{(j,t)}$, $D \cdot X = \sum_{i=0}^N \sum_{j=1}^N \sum_{t=1}^T d_{i,j,t} x_{i,j,t}$ represents the overall route length, and $d_{i,j,t} x_{i,j,t}$ represents the the route length between *node* $v_{(i,t-1)}$ and $v_{(j,t)}$. The overall route length is no longer than a threshold μ , and the sub-route length is no longer than a threshold λ , satisfying the route length constraints.

Solution: this integer programming problem can be solved by various techniques, such as cutting plane, branch and bound, and heuristic search [51]. The basic steps include narrowing the solution space, finding integer-feasible solutions, and discarding space without better integer-feasible solutions. In this work, we employ the Integer Linear Programming Solver from the Matlab Optimization Toolbox⁷ to find the optimal solution.

7 EVALUATION

In this section, we evaluate the performance of our framework based on two real-world traffic violation datasets from two Chinese cities (Chengdu and Xiamen. We first describe the datasets and the experiment settings. Then, we present the evaluation results and runtime performance. Finally, we give several case studies.

7. <https://www.mathworks.com/help/optim/index.html>.

7.1 Experiment Setting

7.1.1 Datasets

We evaluate our framework in Xiamen and Chengdu in China. Xiamen is a coastal city, and we obtain its traffic violations and various road environment datasets during September 2016. Also, we obtain its bike returning data from June to December in 2017. Chengdu is an inland city, and we collect the data from a part of its second ring road area during November 2016. The data analyses of the spatiotemporal contexts and bike returning data can be found at Appendix 4. The traffic violation data and road environment data are summarized in Table 1 and the details and preprocessing steps are elaborated as follows.

Traffic violation data: The traffic violation data are obtained through analyzing crowd behaviors of drivers⁸, consisting of three categories, i.e., illegal turning, illegal parking, and speeding, detailed in Table 1. Specifically, since the data from Xiamen and Chengdu are both in one month, the date and time features exclude *quarter* and *month of a year*. Fig. 6 shows the temporal distributions of traffic violations in Chengdu (November) and Xiamen (September), respectively. We can find that traffic violations in Chengdu mainly happen in the daytime while those in Chengdu mainly happen at midnight. The number of traffic violations in Chengdu fluctuated regularly during November, 2016, while that in Xiamen was influenced by two typhoons (i.e., Meranti on September 15th and Megi on September 27th) in September. Fig. 7 shows the temporal distributions of traffic violations in a typical weekday and weekend Chengdu and Xiamen, from which we can find that the distributions of weekdays and weekends in Chengdu are similar, while in Xiamen, there are two traffic violation peaks in weekends and one traffic violation peak in weekdays.

Road environment data: After clustering traffic violations into traffic violation-prone locations, for each location, we collect its corresponding road environment data, i.e., POI, road category, speed limit and meteorological data, as shown in Table 1. For each location, we count the number

8. More details about the traffic violation crowdsensing can be found at [7], and the traffic violation dataset can be found at https://github.com/zhihanjiang/traffic_violations.

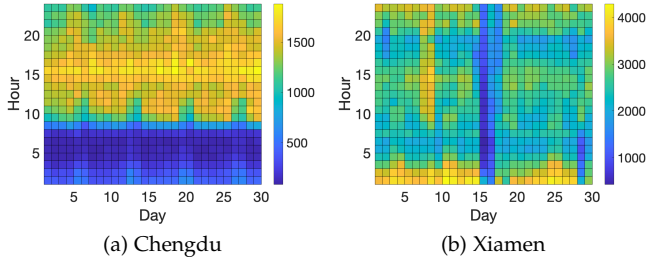


Fig. 6. The traffic violation number in a month.

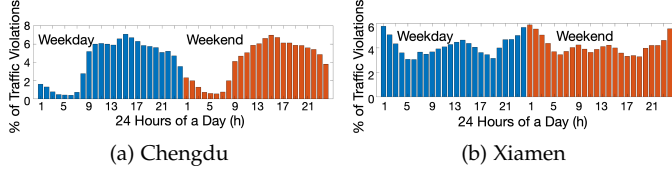


Fig. 7. The temporal distributions of traffic violations in a typical weekday and weekend.

of five different categories of POIs, *business, accommodation, entertainment, infrastructure, tourism* within 200 meters as its POI features, and we get its road category and speed limit from Gaode Map Open Platform. Note that for road intersections, if it is in the intersection of roads with different road categories and speed limits, its road category and speed limit are defined as the lowest road category and the lowest speed limit. We also compiled an hourly meteorology dataset for each traffic violation hotspots, containing the *temperature, dew point, humidity, wind speed, and pressure*, based on the data from the Weather Underground API.

Bike returning data: The bike returning data are from Meituan Bike (Mobike), one of the largest bike-sharing system in China, including the location and time of 23,173,453 bike returning events in Xiamen from June to December in 2017. We use this dataset to extract weekly spatiotemporal patterns of bike returning events of traffic violation-prone locations to estimate potential voluntary participants.

7.1.2 Evaluation Plan

We evaluated the performance of our framework in an online manner. We first cluster traffic violations into traffic violation-prone locations, and start training the CSTA model with a small labeled sample collected in the first week. Then we infer labels for the unlabeled data in first week with the learned CSTA model and get a labeled dataset with a time window of one week. Then we update the training dataset and the model with data collected from a newer time window. For patrol route optimization, we generated the optimal patrol routes twice every day (8:00-12:00 and 14:00-18:00) for each district in each city, and compare the relative coverage ratio of traffic violation hotspots and the average route length with the baseline methods to evaluate the effectiveness of our approach. For traffic police, the origins and destinations of their patrol routes are the same, i.e., $v_{(0,0)}$ is equal to $v_{(N,T)}$. In Chengdu, we only schedule patrol routes for traffic police, and in Xiamen, we further

take the potential voluntary participants into consideration leveraging bike sharing data.

For each traffic violation-prone location, we aggregate its traffic violations hourly and divide its violation level of each hour into three categories, $\{0, 1, 2\}$. More specifically, we first calculate the mean (m_t) and standard deviation (std_t) of traffic violation values in hour t . Then, we define the violation level of location i in hour t as follows:

$$Level_{(i,t)} = \begin{cases} 0, & \text{if } tv_{(i,t)} \leq m_t \\ 1, & \text{if } m_t < tv_{(i,t)} \leq m_t + 2std_t \\ 2, & \text{if } tv_{(i,t)} > m_t + 2std_t \end{cases}$$

where $tv_{(i,t)}$ is the number of traffic violations of location i in hour t . If the number of violations in a location is lower than the average number of violations of all locations at this time, its violation level is zero, which means it is not a violation hotspot. For traffic violation hotspots, their violation levels are divided into two categories, i.e., $\{1, 2\}$.

7.1.3 Evaluation Metric

We compared the results of traffic violation hotspot inference with the ground truth dataset to evaluate the accuracy of a multi-classification method. The metrics used for traffic violation hotspot inference are *precision, recall, and F1-Score*.

Based on the definitions in Section 6, we compared the patrol routes' relative coverage ratio cov_r , average route length of the proposed patrol routes len , and the maximal length of sub-routes $slen$ of the proposed TILP method with baseline methods to evaluate its effectiveness:

$$cov_p = \frac{\sum_{i=1}^k \sum_{t=1}^{T-1} Level_{(q_t,t)}^i}{\sum_{n=1}^N \sum_{t=1}^{T-1} Level_{(n,t)}} \quad (12)$$

$$cov_b = \frac{\sum_{t=1}^{T-1} \sum_u p_{(u,t)}^{crowd} Level_{(u,t)}}{\sum_{n=1}^N \sum_{t=1}^{T-1} Level_{(n,t)}} \quad (13)$$

$$cov_{max} = \frac{\sum_{i=1}^k \sum_{t=1}^{T-1} \max_{i_{th}} \{Level_{(j,t)}^i | 1 \leq j \leq n\}}{\sum_{n=1}^N \sum_{t=1}^{T-1} Level_{(n,t)}} \quad (14)$$

$$cov_r = \frac{cov_p + cov_b}{cov_{max}}, \quad len = \frac{\sum_{i=1}^k l_i}{k} \quad (15)$$

$$slen = \max(\{d_{i,j,t} x_{i,j,t} | i, j \in [0, N], t \in [1, T]\}) \quad (16)$$

The cov_p means the ratio of traffic violation hotspots covered in patrol routes of the first group of patrol participants to all traffic violation hotspots during working hours, and the traffic violation hotspots are weighted by their violation levels. The cov_b means the expectation of traffic violation hotspots covered by voluntary participants, where u denotes the set of nodes uncovered in the patrol routes of the first group of patrol participants. If the voluntary participants are out of consideration, cov_b is set to 0. The cov_{max} means the maximal ratio of weighted traffic violation hotspots that can be covered by limited patrol resources, and $\max_{i_{th}} \{Level_{(j,t)}^i | 1 \leq j \leq n\}$ means the i_{th} maximal $Level$ at time t . The relative coverage ratio cov_r is the ratio of the coverage ratio of traffic violation

hotspots to the maximal coverage ratio that can be achieved with limited patrol resources. The len means the average length of all patrol routes during working hours. If the cov_r of a method is higher than other methods, it means this method can cover more traffic violation hotspots weighted by their levels than other methods. If the len of a method is lower than other methods, it means this method can generate patrol routes with lower travel costs. $slen$ is the maximal value of lengths of sub-routes in patrol routes. To guarantee timely transfer from one patrol node to another, $slen$ should not be too long, and if the $slen$ of a method is lower than other methods, it means the longest time this method requires to transfer from one patrol node to another is shorter than that of other methods.

7.1.4 Baseline Methods

We compare our method with various baseline methods. For traffic violation hotspot classification, the dataset is split into labeled dataset D_L^{all} (80%) and unlabeled dataset D_U^{all} (20%). we compared our CSTA model with the following baselines:

STGCN [17]: STGCN (Spatio-Temporal Graph Convolutional Network) is a deep learning framework to forecast time series in traffic domain. Through spatiotemporal convolutional blocks, it combines graph convolution with gated temporal convolution. The problem is formulated on graphs. Nodes of the graph are traffic violation-prone locations, and the weights of edges are calculated using geographical distances. For each node, it has spatial features (i.e., the extracted spatial contextual features) and temporal features (i.e., the time series of traffic violations in this location). The time series are divided into the two parts, i.e., the first 80% for training and the remaining 20% for testing.

DCRNN [18]: DCRNN (Diffusion Convolutional Recurrent Neural Network) is a deep learning framework leveraging the encoder-decoder architecture and the scheduled sampling technique to facilitate long-term traffic forecasting. Similar to STGCN, the problem is formulated on graphs, and the spatiotemporal features is the same as that for STGCN. Nodes of the graph are traffic violation-prone locations, and the weights of edges are calculated using the Pearson correlation coefficients of temporal features [18].

AGCRN [52]: AGCRN (Adaptive Graph Convolutional Recurrent Network) captures fine-grained correlations in traffic series automatically for traffic forecasting. The problem is formulated on graphs based on the traffic violation series similarity, and the nodes of the graph are traffic violation-prone locations.

CLarge: it directly trains a XGBoost classifier based on D_L^{all} leveraging the spatiotemporal contextual features, and the performance is evaluated based on D_U^{all} .

Csmall: it directly trains a XGBoost classifier with the small labeled dataset D_L^0 sampled from the first week in D_L^{all} leveraging the spatiotemporal contextual features and the performance is evaluated based on D_U^{all} .

CSAL: it exploits active learning to train the classifier iteratively from the small labeled dataset D_L^0 sampled from the first week in D_L^{all} leveraging the spatiotemporal contextual features, and generate queries for crowd automatically. Its performance is evaluated based on D_U^{all} .

CSTT: it first train three classifiers from the small labeled dataset D_L^0 . These classifiers are then iteratively refined

using unlabeled data in the tri-training process. The final labels of the unlabeled data are determined by majority voting. Its performance is evaluated based on D_U^{all} .

In order to achieve fair comparison, we make sure that the initial labeled dataset D_L^0 used in each semi-supervised-learning-based methods (including SMALL, CSAL, CSTT, and CSTA) are the same, and the total number of queries generated in each active learning-based methods (including CSAL and CSTA) are the same. Besides, the parameters used in XGBoost classifiers in all methods are the same. For deep-learning-based methods (STGCN, DCRNN, and AGCRN), the training set and testing set are the same.

For patrol route optimization, we compared our graph-based model with the following baselines:

GRASP [53]: a Greedy Randomized Adaptive Search algorithm. At each iteration of the GRASP, a feasible solution is constructed by a greedy randomized algorithm. At each step of the construction phase, next checkpoint to be visited is randomly drawn according to distance based probability. The above procedures are repeated until all routes are closed.

DGS [54]: a Dynamic Greedy Solution for patrol route planning. In this baseline method, we greedily select the locations with highest violation level for each patrol route, and guarantee the patrol routes are vertex-disjoint. The the importance of each patrol node in this algorithm is determined by its traffic violation level.

YEN [55]: in this baseline method, we use the algorithm proposed by Yen to find k -shortest paths, and the edge weight from one location to another is determined by the route distance.

TS-MOEA [56]: a two-stage evolutionary algorithm for multi-objective multi-depot vehicle routing problem with time windows (MDVRPTW). It first finds extreme solutions and forms a coarse Pareto front, and then it extends the found extreme solutions for approximating the whole Pareto front. We define the traffic police and violation hotspots as the vehicles and customers in MDVRPTW, respectively.

CCMO [57]: the CCMO method evolves one population to solve the original routing problem with time windows, and evolves another population to solve a helper problem derived from the original one. The problem definitions are similar to TS-MOEA.

In order to achieve fair comparison, we make sure that the traffic violation hotspot distribution, route distances between traffic violation-prone locations, and the number of patrol routes to be optimized are the same for each method.

7.2 Evaluation Results

7.2.1 Traffic Violation-Prone Location Extraction

After traffic violation-prone location extraction, we get 84 traffic violation-prone locations in Chengdu and 150 in Xiamen. For 84 locations in Chengdu, 33 of them are from Jinniu District. 29 of them are from Qingyang District. 6 of them are from Jinjiang District, and 16 of them are from Chenghua District. For 150 locations in Xiamen, 60 of them are from Huli District. 86 of them are from Siming District, and only 4 of them are from other districts.

Therefore, we focus on Jinniu District, Qingyang District and Chenghua District in Chengdu, and Siming District

and Huli District in Xiamen. For each traffic violation-prone location, we aggregate its traffic violations hourly, and determine its hourly violation level during working hours (8:00-12:00 and 14:00-18:00). Finally, we get 18,720 instances in Chengdu (12,594, 5638, 488 for category 0, 1, 2, respectively) and 35,040 instances in Xiamen (26,062, 7,664, 1,314 for category 0, 1, 2, respectively).

7.2.2 Traffic Violation Hotspot Inference

Traffic violation series contain sudden changes and are influenced by spatiotemporal contexts, and the experiment results (Table 2) show the effectiveness of using context-awareness scheme to infer traffic violation hotspots. For **AGCRN** method, it directly captures the correlations in traffic violation series without the contextual features for traffic forecasting. The **STGCN** and **DCRNN** methods take the spatial contextual features of the nodes into consideration, outperforming the **AGCRN** method. However, as for the temporal features, they directly use the traffic violation time series without temporal contextual factors. The deep-learning-based methods (**AGCRN**, **STGCN**, and **DCRNN**) are more suitable for forecasting time series with kind of self-regularity attribution.

After incorporating the both of the spatial and temporal contextual features, the **CLarge** method achieves good performance on D_U^{all} using the large labeled dataset D_L^{all} , outperforming **STGCN**, **DCRNN**, and **AGCRN**. Also, **CLarge** is more efficient than **STGCN**, **DCRNN**, and **AGCRN**. It cost 6.42 and 15.29 seconds to finish training in Chengdu and Xiamen, respectively, while **STGCN** cost 108.22 and 180.83 seconds. **DCRNN** cost 42.59 and 54.89 minutes, and **AGCRN** cost 47.48 and 50.76 seconds.

However, in practice, collecting such a training set is time-consuming (the large labeled dataset is collected in one month) and hinders the timely optimization for patrol routes. Therefore, we further incorporate active learning and tri-training into the model. According to observations, the number of patrol teams for each district is around 10. Therefore, we set the label ability of crowd as 10 every day for every station. Since the **CSTA** model is updated every week, the crowd for every district are going to label 70 instances in the ambiguous set A every week. Therefore, the total number of labeling queries dealt with by crowd in each active learning-based methods (including **CSAL** and **CSTA**) are set to 840 and 560 from three districts in Chengdu and two districts in Xiamen, respectively.

Based on the value of ϵ discussed above, we study the $F1$ -score of the **CSTA** method against various $|D_L^0|$ values. As shown in Fig. 8, the initial training sets of size 1900 and 1100 are large enough to obtain the $F1$ -score higher than 90% in Chengdu and Xiamen, respectively. Therefore, we select $|D_L^0| = 1900$ for Chengdu and $|D_L^0| = 1100$ for Xiamen as the optimal initial training set sizes in the following experiments. Therefore, in Chengdu, we select 1900 labels from the first week of D_L^{all} as the initial label set, and limit the total number of labels to 2740 for active-learning-based methods. In Xiamen, we select 1100 labels from the first week of D_L^{all} as the initial label set, and limit the total number of labels to 1660 for active-learning-based methods.

As shown in Table 2, the **CSmall** baseline does not perform well, especially in *recall* and *F1-score*, and the

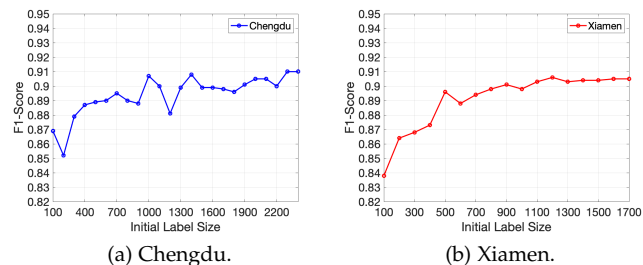


Fig. 8. Initial label size selection in Chengdu and Xiamen.

TABLE 2
Traffic violation hotspot inference results.

| City | Chengdu | | | Xiamen | | | |
|------|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Methods | P | R | F1 | P | R | F1 |
| | STGCN [17] | 0.819 | 0.812 | 0.815 | 0.889 | 0.885 | 0.887 |
| | DCRNN [18] | 0.793 | 0.786 | 0.789 | 0.869 | 0.870 | 0.870 |
| | AGCRN [52] | 0.772 | 0.858 | 0.806 | 0.775 | 0.900 | 0.821 |
| | CLarge | 0.895 | 0.860 | 0.876 | 0.897 | 0.880 | 0.888 |
| | CSmall | 0.849 | 0.790 | 0.814 | 0.872 | 0.736 | 0.785 |
| | CSAL | 0.908 | 0.805 | 0.845 | 0.881 | 0.790 | 0.823 |
| | CSTT | 0.870 | 0.825 | 0.845 | 0.883 | 0.793 | 0.830 |
| | CSTA | 0.915 | 0.888 | 0.901 | 0.920 | 0.892 | 0.906 |

P is the *precision*, R is the *recall*, and F1 is the *F1-Score*.

CSAL and **CSTT** methods perform better than **CSmall** by exploiting active learning and tri-training, respectively. The proposed **CSTA** method further improves the performance by combing active learning and tri-training, achieving an $F1$ -score of 0.901 with *precision* = 0.915 and *recall* = 0.888 using only 18.30% of labels in Chengdu, and achieving an $F1$ -score of 0.906 with *precision* = 0.920 and *recall* = 0.892 using only 5.92% of labels in Xiamen.

7.2.3 Patrol Route Optimization

We present the evaluation results on patrol route optimization in Table 3. Based on observations, we set the number of traffic routes for each police station as 10, set μ as 10×15 km for Siming and Qingyang Districts and as 10×10 km for other districts. Besides, since the traffic police should move from one node to another within 10 minutes and its average speed is 60 km/h, the sub-route length threshold λ is set to 10 km. The **TILP+Bike** method in Xiamen incorporate the crowd tensor B calculated using the bike sharing data, while the **TILP** does not take the voluntary participants into account (all elements in B are set to 0). The evaluation results are the average results of all patrol routes generated in a month for each district. To eliminate the influence of randomness in **GRASP** method, we repeat this baseline method 100 times and get its average performance.

The **GRASP** method does not perform well in route length (**len**), sub-route length (**slen**), and relative coverage ratio (**cov_r**). Among all methods without voluntary participants, the **DGS** baseline method achieves the best performance in **cov_r**, but it does not perform well in **len** and **slen**, while the **YEN** baseline achieves the best performance in **len** and **slen** but the worst in **cov_r**. Although

TABLE 3
Patrol route optimization results.

| District | Chenghua (Chengdu) | | | Qingyang (Chengdu) | | | Jinniu (Chengdu) | | | Siming (Xiamen) | | | Huli (Xiamen) | | |
|--------------|--------------------|------------------|------|--------------------|------------------|------|------------------|------------------|-------|-----------------|------------------|-------|---------------|------------------|-------|
| | Methods | cov _r | len | slen | cov _r | len | slen | cov _r | len | slen | cov _r | len | slen | cov _r | len |
| GRASP [53] | 48.87 | 16.53 | 9.42 | 28.19 | 20.60 | 8.99 | 27.01 | 17.88 | 13.87 | 15.60 | 36.67 | 19.65 | 22.43 | 25.03 | 17.29 |
| DGS [54] | 78.72 | 12.72 | 9.25 | 81.20 | 18.69 | 8.99 | 80.28 | 14.34 | 10.95 | 80.58 | 30.28 | 18.07 | 80.07 | 22.22 | 15.86 |
| YEN [55] | 16.51 | 3.49 | 1.78 | 0.47 | 11.14 | 5.59 | 1.42 | 0.91 | 0.46 | 5.43 | 2.43 | 1.49 | 18.25 | 4.00 | 2.00 |
| TS-MOEA [56] | 35.45 | 6.81 | 3.63 | 24.06 | 14.70 | 7.09 | 12.54 | 3.91 | 1.68 | 7.90 | 10.53 | 6.08 | 7.20 | 5.55 | 2.42 |
| CCMO [57] | 52.34 | 4.67 | 3.73 | 29.86 | 12.97 | 7.26 | 15.67 | 2.34 | 2.29 | 11.88 | 7.75 | 6.97 | 15.20 | 4.24 | 2.65 |
| TILP | 78.72 | 9.74 | 9.42 | 80.09 | 14.86 | 8.16 | 80.07 | 9.53 | 9.28 | 76.16 | 14.62 | 10.00 | 80.07 | 9.42 | 9.98 |
| TILP+Bike | - | - | - | - | - | - | - | - | - | 87.37 | 14.95 | 9.92 | 85.67 | 9.93 | 8.71 |

the TS-MOEA method performs well in **len** and **slen**, it achieves poorer **cov_r** than other methods except for **YEN**, especially in Xiamen. The **CCMO** method outperform **TS-MOEA** with higher **cov_r** and lower **len** and **slen**. However, it emphasizes on minimizing route length with limited resources, while the patrol route optimization emphasizes on the effectiveness of patrolling with reasonable route length. Compared with the **CCMO** method, the proposed **TILP** method achieves much higher **cov_r** at the cost of increasing route lengths within reasonable ranges, which is more practical in real-world application scenarios. Also, compared with the **DGS** method, the proposed **TILP** method achieves comparable relative coverage ratio while reduce the average route length to 68.23% and 45.34% and reduce the sub-route length to 91.56% and 59.13% in Chengdu and Xiamen, respectively. Besides, since the lengths of overall routes and sub-patrol routes are limited, the differences of travel time among patrol teams would be limited, guaranteeing the balance among patrol routes.

Furthermore, we take potential voluntary participants into consideration in Xiamen (the **TILP+Bike** in Table 3). The distance threshold τ of the spatiotemporal pattern extraction for bike returning events is set to 100 meters, and the probability of rejecting rejecting the patrol task p_{rej} is set to 0.9. According to the experiment results, we can find that through leveraging the crowd power, the **TILP+Bike** achieves the best performance in relative coverage ratio.

7.2.4 Runtime Performance

We deployed our framework on a server with NVIDIA GeForce GTX 1080 Ti 11GB and 32GB RAM. The framework works in an online manner and is updated every week. In Chengdu and Xiamen, it takes an average of 572.87 seconds and 248.40 seconds to cluster one round of scattered traffic violations into traffic violation-prone locations, an average of 72.70 seconds and 554.06 seconds to update one round of CSTA models, and an average of 0.61 seconds and 5.61 seconds to allocate one round of patrol tasks, respectively. Therefore, the average time of processing new data collected every week is about 10.77 minutes in Chengdu and 13.47 minutes in Xiamen.

7.3 Case Studies

We developed a real-world Traffic Violation Hotspot Patrolling Information Service System⁹ as shown in Fig. 9,

9. <https://research.spatial-crowdsensing.com/driving-violation/>

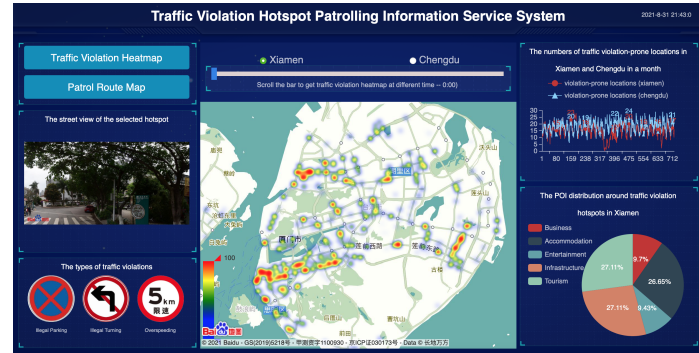


Fig. 9. The developed Traffic Violation Hotspot Patrolling Information Service System.

and conducted case studies on traffic violation hotspots in Xiamen and Chengdu, respectively.

7.3.1 Traffic Violation Hotspots in Chengdu

Fig. 10a shows three traffic violation hotspots successfully added to patrol routes by our framework in Chenghua District, Jinniu District, and Qingyang District respectively. The traffic violation hotspot in Chenghua District is the 2nd Ring Road North 4th Segment between Jiefang Road and Fuqing Road. This road segment has a speed limit of 60 km/h, while many drivers exceed the speed limit there, and thus it requires attentions. The traffic violation hotspot in Jinniu District is the road intersection of 1st Ring Road and People North Road. This is a busy road intersection near to a big shopping mall (Wanda Plaza), and thus a lot of illegal parking behaviors happen there. Therefore it is frequently added to patrol routes. The traffic violation hotspot in Qingyang District is the road intersection of Twelve Bridge Road and Xi'an South Road. The left turn from Xi'an South Road to Twelve Bridge Road is not allowed in this intersection, but many drivers still turn left there. We even found a car was turning left in this intersection in the street view picture.

7.3.2 Traffic Violation Hotspots in Xiamen

Fig. 10b shows two traffic violation hotspots successfully added to patrol routes by our framework in Huli District and Siming District respectively. The traffic violation hotspot in Huli District is a road segment of Jiahe Road between Hubin North Road and Songbai Road. Jiahe Road is a main road across Xiamen Island, and this road segment of

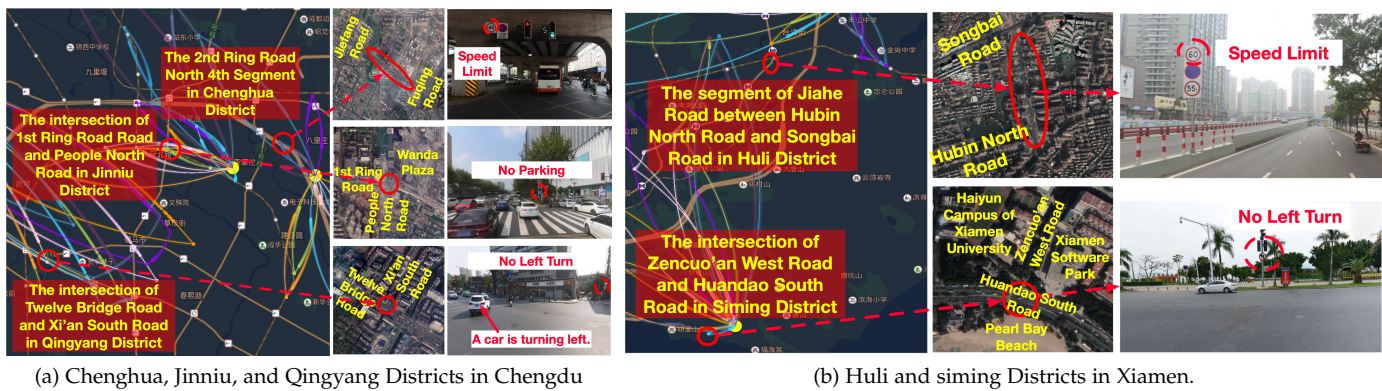


Fig. 10. The traffic violation hotspots frequently added to patrol routes.

Jiahe Road has a speed limit of 60 km/h. However, speeding frequently happens on this road segment, and thus patrol participants should pay more attention to this segment. The traffic violation hotspot in Siming District is the intersection of Zencuo'an West Road and Huandao South Road, which is a busy road intersection near to the Haiyun campus of Xiamen University, Xiamen Software Park, and a tourist hotspot (Pearl Bay Beach). Illegal left turning behaviors from Zencuo'an West Road to Huandao South Road frequently happen in this intersection. Therefore, it is reasonable to add this intersection to patrol routes.

8 DISCUSSION

We discuss the following limitations of our work.

(1) Updating Hotspots for Emergencies. The proposed framework is adaptive to gradual trends of dynamic traffic violation hotspots and design routine patrol routes accordingly. However, it is currently unable to quickly respond for emergencies. In real-world settings, road emergencies such as traffic accidents are usually reported quickly to traffic police. Therefore, in the future, in addition to the inferred traffic violation hotspots, we can also update hotspots for the crowd-reported emergencies in patrol route optimization. More specifically, if an emergency occurs in the road network, it can be directly assigned to the nearest patrol participant so as to guarantee timely response, and then re-optimize the sub-tensor of the adjacency tensor X representing the following patrol routes based on the current distribution of patrol participants and traffic violation hotspots. In this way, the framework can be adapted to emergencies.

(2) Designing Incentive Mechanism for Voluntary Participants. Currently, the potential voluntary participants for traffic violation hotspots is simply estimated leveraging the spatiotemporal patterns of bike returning events nearby. However, in real-world settings, the probability of successfully recruiting crowd volunteers is influenced by various factors such as weather conditions, rewards, and participants' profiles. For example, the crowd patrol ability may decrease under appalling weather conditions, and the crowd in residential districts may be more likely to receive the crowdsensing tasks than those in business districts. In the future, we plan to incorporate more impact factors when building the crowd tensor B , and design a reward mechanism to further encourage crowdsensing participants.

(3) Further Applications in Mobile Computing. The proposed framework can be applied to various domains other than transportation. For example, it can be applied to infer crowded nodes in the mobile network and schedule edge devices to share the traffic in those crowded nodes so as to increase the utility and service quality [58]. It can also be applied to find specific spatiotemporal contexts with high error risk for software and hardware and infer future error hotspots, and schedule crowd intelligence to verify the inferred hotspots. Besides, the framework is compatible with many other application scenarios in cities, such as crime area patrolling and public facility maintenance.

9 CONCLUSION

In this paper, we propose a mobile crowdsensing framework for dynamic traffic violation hotspot inference and adaptive patrol task scheduling. Specifically, we first extract traffic violation events from heterogeneous crowd-sensed data, and cluster them into traffic violation-prone locations with a road network-based clustering algorithm. Then, we propose a context-aware self-adaptive learning approach to infer traffic violation hotspots. Finally, we find the patrol routes with a maximal coverage ratio of traffic violation hotspots under resource constraints by a proposed tensor-based integer linear programming modeling method. We evaluate our framework using real-world datasets collected from Xiamen and Chengdu. Results show that our framework accurately infers traffic violation hotspots and generates patrol routes adaptively, outperforming the baseline methods.

In the future, we intend to improve this work from the following aspects. First, we plan to incorporate traffic accident records, and collect more traffic violation data leveraging crowdsensing platforms. Second, we plan to consider more factors in patrol route optimization, for example, the influence of incentives on patrol participants. Third, we plan to evaluate our framework in more cities with different geographic and meteorological conditions, and explore the similarities and differences of relative spatiotemporal contextual features in different cities.

10 ACKNOWLEDGMENTS

We would like to thank the reviewers for their constructive suggestions. This research is supported by NSF of China No. 61802325, No. 62102349, No. 61872306, Fundamental

Research Funds for the Central Universities (20720200031), and the Research Grants Council of the Hong Kong Special Administrative Region, China under Grant No. C6030-18G.

REFERENCES

- [1] G. Zhang, K. K. Yau, and G. Chen, "Risk factors associated with traffic violations and accident severity in China," *Accident Analysis & Prevention*, vol. 59, pp. 18–25, 2013.
- [2] M. O. o. O. New York City, "Vision Zero Year Five Report," New York, Tech. Rep., 2019.
- [3] "World health organization: Road traffic injuries fact sheet," [EB/OL], <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries> Accessed April 29, 2020.
- [4] G. DeAngelo and B. Hansen, "Life and death in the fast lane: Police enforcement and traffic fatalities," *American Economic Journal: Economic Policy*, vol. 6, no. 2, pp. 231–57, 2014.
- [5] X. Chen, "Fast patrol route planning in dynamic environments," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 4, pp. 894–904, 2012.
- [6] D. Zhang, B. Guo, B. Li, and Z. Yu, "Extracting social and community intelligence from digital footprints: an emerging research area," in *International Conference on Ubiquitous Intelligence and Computing*. Springer, 2010, pp. 4–18.
- [7] Z. Jiang, L. Chen, B. Zhou, J. Huang, T. Xie, X. Fan, and C. Wang, "itv: Inferring traffic violation-prone locations with vehicle trajectories and road environment data," *IEEE Systems Journal*, pp. 1–12, 2020.
- [8] Y. Li, M. Abdel-Aty, J. Yuan, Z. Cheng, and J. Lu, "Analyzing traffic violation behavior at urban intersections: A spatio-temporal kernel density estimation approach using automated enforcement system data," *Accident Analysis & Prevention*, vol. 141, p. 105509, 2020.
- [9] J. Gao and K. Ozbayb, "Hotspot identification, ranking and impact estimation of illegal parking using spatial association and queueing model," 2020.
- [10] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 16–28, 2017.
- [11] B. Guo, H. Chen, Z. Yu, X. Xie, S. Huangfu, and D. Zhang, "Fliermeet: a mobile crowdsensing system for cross-space public information reposting, tagging, and sharing," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2020–2033, 2014.
- [12] A. Farshad, M. K. Marina, and F. Garcia, "Urban wifi characterization via mobile crowdsensing," in *2014 IEEE Network Operations and Management Symposium (NOMS)*. IEEE, 2014, pp. 1–9.
- [13] X. Li and D. W. Goldberg, "Toward a mobile crowdsensing system for road surface assessment," *Computers, Environment and Urban Systems*, vol. 69, pp. 51–62, 2018.
- [14] Z. Jiang, Y. Liu, X. Fan, C. Wang, J. Li, and L. Chen, "Understanding urban structures and crowd dynamics leveraging large-scale vehicle mobility data," *Frontiers of Computer Science*, vol. 14, no. 5, pp. 1–12, 2020.
- [15] J. Hong, J. Park, G. Lee, and D. Park, "Endogenous commercial driver's traffic violations and freight truck-involved crashes on mainlines of expressway," *Accident Analysis & Prevention*, vol. 131, pp. 327–335, 2019.
- [16] E. Roidl, B. Frehse, and R. Höger, "Emotional states of drivers and the impact on speed, acceleration and traffic violations—a simulator study," *Accident Analysis & Prevention*, vol. 70, pp. 282–292, 2014.
- [17] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *IJCAI*, 2018.
- [18] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.
- [19] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, Nov. 2011.
- [20] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4W1H in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 42–48, Aug. 2014.
- [21] C. Chen, D. Zhang, X. Ma, B. Guo, L. Wang, Y. Wang, and E. Sha, "Crowddeliver: Planning City-Wide Package Delivery Paths Leveraging the Crowd of Taxis," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–19, 2016.
- [22] E. Wang, M. Zhang, X. Cheng, Y. Yang, W. Liu, H. Yu, L. Wang, and J. Zhang, "Deep learning-enabled sparse industrial crowdsensing and prediction," *IEEE Transactions on Industrial Informatics*, 2020.
- [23] S. Tarapiah, S. Atalla, and R. AbuHania, "Smart on-board transportation management system using gps/gsm/gprs technologies to reduce traffic violation in developing countries," *International Journal of Digital Information and Wireless Communications (IJDIWC)*, vol. 3, no. 4, pp. 96–105, 2013.
- [24] N. Aliane, J. Fernandez, M. Mata, and S. Bemposta, "A system for traffic violation detection," *Sensors*, vol. 14, no. 11, pp. 22 113–22 127, 2014.
- [25] A. C. P. Uy, A. R. F. Quiros, R. A. Bedruz, A. Abad, A. Bandala, E. Sybingco, and E. P. Dadios, "Automated traffic violation apprehension system using genetic algorithm and artificial neural network," in *2016 IEEE Region 10 Conference (TENCON)*. IEEE, 2016, pp. 2094–2099.
- [26] H. R. Mampilayil and K. Rahamathullah, "Deep learning based detection of one way traffic rule violation of three wheeler vehicles," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE, 2019, pp. 1453–1457.
- [27] G. Chen and D. Kotz, "A survey of context-aware mobile computing research," *Dartmouth Computer Science Technical Report TR2000-381*, 2000.
- [28] L. Chen, X. Fan, L. Wang, D. Zhang, Z. Yu, J. Li, T.-M.-T. Nguyen, G. Pan, and C. Wang, "Radar: Road obstacle identification for disaster response leveraging cross-domain urban data," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–23, 2018.
- [29] L. Allen, T. Miki, and S. Shoab, "Context aware search service," Jul. 17 2003, uS Patent App. 10/134,814.
- [30] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.
- [31] T. Qian, B. Liu, L. Chen, and Z. Peng, "Tri-training for authorship attribution with limited training data," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 345–351.
- [32] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR.org, 2017, pp. 2988–2997.
- [33] Y. Zhang, J. Wen, X. Wang, and Z. Jiang, "Semi-supervised learning combining co-training with active learning," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2372–2378, 2014.
- [34] N. Rubens, M. Elahi, M. Sugiyama, and D. Kaplan, "Active learning in recommender systems," in *Recommender systems handbook*. Springer, 2015, pp. 809–846.
- [35] L. Wang, D. Zhang, A. Pathak, C. Chen, H. Xiong, D. Yang, and Y. Wang, "Ccs-ta: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, 2015, pp. 683–694.
- [36] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Proceedings. IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004.(IAT 2004)*. IEEE, 2004, pp. 302–308.
- [37] W. Hoffman and R. Pavley, "A method for the solution of the n th best path problem," *Journal of the ACM (JACM)*, vol. 6, no. 4, pp. 506–514, 1959.
- [38] T. Eilam-Tzoref, "The disjoint shortest paths problem," *Discrete applied mathematics*, vol. 85, no. 2, pp. 113–138, 1998.
- [39] S. Tragoudas and Y. L. Varol, "Computing disjoint paths with length constraints," in *International Workshop on Graph-Theoretic Concepts in Computer Science*. Springer, 1996, pp. 375–389.
- [40] N. Ahituv and O. Berman, *Operations management of distributed service networks: A practical quantitative approach*. Springer Science & Business Media, 2012.
- [41] L. Hajibabai and D. Saha, "Patrol route planning for incident response vehicles under dispatching station scenarios," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 1, pp. 58–70, 2019.
- [42] A. Mukhopadhyay and Y. Vorobeychik, "Prioritized allocation of emergency responders based on a continuous-time incident prediction model," in *International Conference on Autonomous Agents and Multi-Agent Systems*, 2017.

[43] M. Dewinter, C. Vandeviver, T. V. Beken, and F. Witlox, "Analysing the police patrol routing problem: A review," *ISPRS International Journal of Geo-Information*, vol. 9, no. 3, p. 157, 2020.

[44] J. Leigh, S. Dunnett, and L. Jackson, "Predictive police patrolling to target hotspots and cover response demand," *Annals of Operations Research*, vol. 283, no. 1-2, pp. 395-410, 2019.

[45] L. Li, Z. Jiang, N. Duan, W. Dong, K. Hu, and W. Sun, "Police patrol service optimization based on the spatial pattern of hotspots," in *Proceedings of 2011 IEEE International Conference on Service Operations, Logistics and Informatics*. IEEE, 2011, pp. 45-50.

[46] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 336-343.

[47] A. Bogomolov, B. Lepri, J. Staiano, N. Oliver, F. Pianesi, and A. Pentland, "Once upon a crime: towards crime prediction from demographics and mobile data," in *Proceedings of the 16th international conference on multimodal interaction*, 2014, pp. 427-434.

[48] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785-794.

[49] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349-360, 2009.

[50] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146-3154.

[51] G. L. Nemhauser and L. A. Wolsey, *Integer Programming and Combinatorial Optimization*, 1988, vol. 20.

[52] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[53] S. Mancini, "A greedy randomized adaptive search for the surveillance patrol vehicle routing problem," in *Recent Developments in Metaheuristics*. Springer, 2018, pp. 305-317.

[54] S. K. Rumi, W. Shao, and F. D. Salim, "Realtime predictive patrolling and routing with mobility and emergency calls data," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, 2020, pp. 964-968.

[55] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712-716, 1971.

[56] J. Wang, T. Weng, and Q. Zhang, "A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2467-2478, 2018.

[57] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 102-116, 2020.

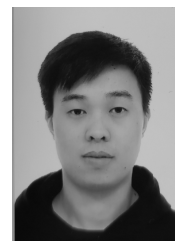
[58] L. Chen, D. Yang, M. Nogueira, C. Wang, D. Zhang *et al.*, "Data-driven c-ran optimization exploiting traffic and mobility dynamics of mobile users," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1773-1788, 2020.



Binbin Zhou received the B.Math. degree from Hangzhou Normal University in 2008, the M.Phil degree in computer science from Hongkong Polytechnic University in 2011, and the Ph.D. degree in computer science from Zhejiang University in 2021. She is currently an Associate Research Fellow with at Zhejiang University City College. Her research interests are mainly in urban computing, big data applications, and ubiquitous computing.



Chenhui Lu received the B.Eng. degree in electronic information engineering from Fujian Normal University, China in 2018. He is currently pursuing the master's degree in Department of Computer Science, Xiamen University. His research interests include Big Data Analytics, Crowdsensing and Urban Computing.



Mingfei Sun received the Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology in 2020. He is currently a research associate at the University of Oxford and jointly a researcher at Microsoft Research Cambridge. Mingfei Sun is generally interested in reinforcement learning theories and applications.



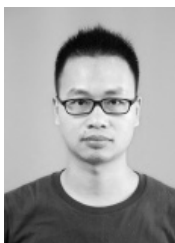
Xiaojuan Ma is an assistant professor of Human-Computer Interaction at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. She received her Ph.D. in Computer Science at Princeton University. She was a post-doctoral researcher at the Human-Computer Interaction Institute of Carnegie Mellon University, and before that a research fellow in the National University of Singapore in the Information Systems department.



Xiaoliang Fan is a Senior Research Specialist at Xiamen University. He received his PhD degree at University Pierre and Marie CURIE, France in 2012. His research interests include Spatio-Temporal Data Mining, Federated Learning, Privacy-aware Computing, etc. He has published 60+ journal and conference papers in these areas. He is a senior member of both IEEE and China Computer Federation (CCF).



Cheng Wang received the Ph.D. degree in information and communication engineering from the National University of Defense Technology, Changsha, China in 2002. He is currently a Professor with Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, China. His research interests include remote sensing image processing, mobile LiDAR data analysis, and multisensor fusion.



Longbiao Chen is an assistant professor with Xiamen University, China. He obtained his Ph.D. degree in computer science from Sorbonne University, France in 2018 and Zhejiang University, China in 2016. His research interests include Mobile Computing, Big Data Analytics, Crowdsensing and Urban Computing.



Zhihan Jiang received the B.E. and M.E. degrees in computer science and technology from Xiamen University, China in 2018 and 2021, respectively. She is currently pursuing the Ph.D. degree in Department of Electrical and Electronic Engineering, The University of Hong Kong. Her research interests include Mobile Computing, Urban Computing, Crowdsensing, and Data Analysis.



Hang Zhu received his B.S. from the Department of Software Engineering, Wuhan University of Technology. He is currently pursuing a M.S. Degree in the Department of Computer Science and Technology at Xiamen University. His research interests focus on Graph Neural Network and Urban Computing.