# Traffic Incident Detection: A Trajectory-based Approach

Xiaolin Han*, Tobias Grubenmann*, Reynold Cheng*, Sze Chun Wong†, Xiaodong Li* and Wenya Sun*
*Department of Computer Science, The University of Hong Kong, Hong Kong SAR, China
†Department of Civil Engineering, The University of Hong Kong, Hong Kong SAR, China
Email: *{xlhan,tobias,ckcheng,xdli,wysun}@cs.hku.hk, †hhecwsc@hku.hk

***Abstract*—Incident detection (ID), or the automatic discovery of anomalies from road traffic data (e.g., road sensor and GPS data), enables emergency actions (e.g., rescuing injured people) to be carried out in a timely fashion. Existing ID solutions based on data mining or machine learning often rely on *dense* traffic data; for instance, sensors installed in highways provide frequent updates of road information. In this paper, we ask the question: Can ID be performed on *sparse* traffic data (e.g., location data obtained from GPS devices equipped on vehicles)? As these data may not be enough to describe the state of the roads involved, they can undermine the effectiveness of existing ID solutions. To tackle this challenge, we borrow an important insight from the transportation area, which uses trajectories (i.e., moving histories of vehicles) to derive *incident patterns*. We study how to obtain incident patterns from trajectories and devise a new solution (called Filter-Discovery-Match (FDM)) to detect anomalies in sparse traffic data. Experiments on a taxi dataset in Hong Kong and a simulated dataset show that FDM is more effective than state-of-the-art ID solutions on sparse traffic data.**

***Index Terms*—Data Mining, Traffic Incident Detection, Sparsity**

## I. Introduction

Advances in traffic data acquisition technologies (e.g., loop sensors, road detectors, Global Positioning System (GPS), and CCTV cameras) provide gigantic amounts of Big Transportation Data (BTD) in real-time. These data (e.g., road traffic conditions, vehicle locations, speeds, pedestrian information) enable urban computing applications (e.g., autonomous vehicles, intelligent navigation, smart traffic light control, and air pollution reduction), with the goal of improving transportation conditions and living quality of citizens. A fundamental problem in BTD is the automatic discovery of incidents (e.g., roadblock and traffic incidents). Particularly, a number of *incident detection* (ID) algorithms have been proposed (e.g., [1]–[4]), which perform mining and machine learning on BTD, identify abnormal traffic states, so that appropriate actions could be taken (e.g., dispatch medical and police resources to prevent life loss, or detour drivers to incident-free routes to avoid congestion).

For existing ID algorithms, the underlying traffic data is often assumed to be *dense* – i.e., each road involved is covered by huge volumes of traffic data (e.g., vehicle locations). This is a reasonable assumption for freeways, where fixed equipment such as road detectors and CCTV cameras are installed to acquire traffic information regularly. In fact, most experiments on existing ID solutions [1]–[4] are conducted on freeways. However, it is doubtful whether these solutions are effective on *urban roads* (i.e., roads in cities with many neighboring junctions and traffic lights), where road detectors may be rare.

An alternative BTD source whose data provide a higher coverage of roads is GPS data. Due to the low costs of GPS devices, they are commonly found in many smart phones and vehicles. They are much less costly than road detectors, and the positions of vehicles can be acquired by GPS devices in real-time. Hence, by using GPS data, the traffic state of urban roads can be obtained. We collected GPS data from the vehicles owned by a taxi company in Hong Kong. A problem common to these data sources is that they may not cover all the roads with the same amount of data. For example, roads located in commercial districts are travelled more frequently than other roads. For roads in residential or suburb areas, traffic data can be *sparse*.

In this paper, we investigate the question: can ID be performed on sparse traffic (e.g., GPS data)? Experiments on GPS data provided by a taxi company show that sparse traffic renders existing ID solutions less effective. The main reason is that these data mining or machine learning solutions often require a huge amount of traffic data over both spatial and temporal dimensions.

In contrast to existing work, we borrow an insight from the transportation community, which makes use of the movement history (or *trajectory*) of a vehicle to derive a *speed pattern* (i.e., an observation of the speed of a vehicle over a certain period of time) [5]. As pointed out by [5], when a vehicle passes through a location where an incident occurs, these speed patterns can be used to detect incidents effectively. Based on this intuition, we develop a solution called Filter-Discovery-Match (**FDM**). The main idea of FDM is to use speed patterns to identify anomalies, in order to find out whether a vehicle is passing the scene of an incident. In sparse settings where roads do not receive a lot of traffic data, FDM yields a lower mean time-to-detect (MTTD) than existing solutions. As FDM requires extensive analysis over trajectory data, the computational cost of our solution can be quite high. To enable fast detection of incidents, we develop exact and approximate detection algorithms. Our experiments on a large taxi dataset in Hong Kong and a simulated dataset show that (1) FDM is more effective than existing ID algorithms, and (2) FDM is computationally efficient.

2375-026X/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDE48307.2020.00190

## II. Related Work

Previous work on traffic incident detection can be divided into three categories: pattern recognition, deviation detection, and machine learning. Most of these categories provide methods for incident detection on freeways. As the data is collected from static detectors installed along freeway roads, these algorithms generally assume that the data is updated at very short time intervals for each individual road segment. This assumption limits their applicability for incident detection on urban roads using GPS-equipped vehicles, as the time between two successive vehicles travelling through some road segments can be very large. We call this the *data sparsity problem in urban roads*.

**Pattern Recognition:** California [6] and DELOS [7] make use of the occupancy difference in the spatial and temporal dimensions to detect incidents. In their experiments, the data is updated in one-minute and thirty-seconds intervals, respectively. Such short intervals are unrealistic for urban roads, however, since many urban road segments are not covered by even one vehicle during some time intervals. To adapt their solution to GPS signals collected by vehicles on urban roads, aggregation over longer time intervals is necessary. This can lead to the problem of a high mean time-to-detect (MTTD).

**Deviation Detection:** Traffic state estimation [8] uses the spatial-temporal feature deviation to estimate the traffic state and to detect incidents using these states. However, the estimated traffic state can be biased when the GPS data is limited. Probabilistic topic modelling [9] uses the Latent Dirichlet Allocation (LDA) equivalent model [10] to calculate the divergence of the current traffic state from the normal traffic state. A traffic state is classified as an incident when the divergence exceeds a user-defined threshold. However, it requires mass data to estimate the real distribution of data. As all of these methods are designed for scenarios with dense datasets, their performance can be negatively affected by the sparsity of GPS data on urban roads.

**Machine Learning:** Neural network (NN) based methods [1], [3] use the upstream and downstream of traffic volume, road occupancy, and traffic speed at different time intervals as features to train NN based models. Support vector machines (SVM) [2], [4] use the same features to detect incidents. Wong and Wong [4] show that SVM models can outperform NN-based models. Convolutional Neural Networks (CNN) [11] are utilized to detect incidents on traffic flow data aggregated over 5 minutes intervals. However, all these methods are designed for freeways and may perform poorly when applied to urban roads, which generate much sparser vehicular data. One could address this problem by aggregating data over a longer time period, but that would significantly increase the MTTD and may render these methods impractical.

## III. Filter-Discovery-Match (FDM)

In this section, we first give definitions of the basic concepts and the problem (Section III-A). Then, we introduce the incident patterns, which are the basic building blocks for incident detection (Section III-B). After that, our FDM method is further explained in Section III-C.

### A. Problem Definition

The core purpose of this paper is to use GPS trajectories for incident detection. A GPS device is continuously tracking its location. Due to technical limitations, the GPS locations are only tracked at discrete timestamps, which results in an approximation of a GPS trajectory using a finite number of GPS points. Each GPS point consists of a timestamp, a location, and the current speed.

The GPS points of two different vehicles usually do not coincide in the same location, even if the two vehicles travel along the same road. Thus, in order to compare the trajectories of two different vehicles, we partition the whole road network into *road segments* and aggregate all the GPS points within a single road segment. The result of this aggregation is the timestamp when the vehicle passed through this segment and the average speed while passing through this segment. Using these average speeds, we can construct a *speed vector* for a sequence of road segments which have been travelled consecutively by a vehicle.

In order to detect incidents, we investigate the speed vectors of affected vehicles. We say that a vehicle has been affected by an incident if it travels through the road segment where the incident happened within the time-interval $[t_{\text{inc}}, t_{\text{inc}}+\tau]$, where $t_{\text{inc}}$ is the time of the incident and $\tau \in \mathbb{R}_+$ is a predefined threshold. We call the speed vector of a vehicles which have been affected by an incident the *incident speed vector*. The incident speed vector has a length $m \in \mathbb{N}$, which means we only consider the last $m$ road segments of the vehicle.

To classify an incident speed vector (i.e., to determine if an incident happened in the corresponding road segment), we need a reference for comparison. In our case, we compare the *incident speed vector* with the historical average speed of the same $m$ road segments. To calculate the average, we only include vehicles which have not been affected by a traffic incident. To account for fluctuations in the traffic behavior, we further consider only vehicles which passsed through the corresponding segments during the same hour or the day and the same day of the week. We call the speed vector of the historical average the *normal speed vector*.

With these definitions, we can now formulate the problem definition of our paper: The aim of traffic incident detection with sparse trajectories is to raise an alarm when an incident happens on a road segment by comparing the speed vector of by-passing vehicles with the corresponding normal speed vector.

### B. Incident Patterns

Experts from the transportation field [5] evaluated the impact of traffic incidents using trajectories, finding that vehicles first reduce their speed, maintain that speed for a certain time-interval, and then finally increase in speed, having passed the incident locations. Based on this finding, our idea is to discover

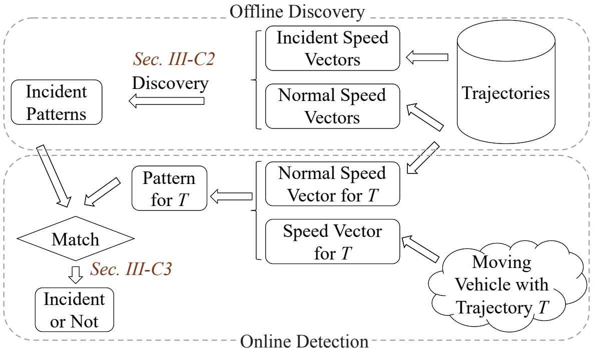

Fig. 1. Overview of **FDM**.

incident patterns by comparing the incident speed vector and the normal speed vector.

For each incident speed vector and normal speed vector we can derive a *candidate speed pattern* by calculating the speed difference for each road segment. Next, we cluster all these candidate speed patterns. Each center of a clusters is an *incident patterns*. If the pattern for a speed vector of a real-time trajectory is close to one of our discovered incident patterns, an alarm is triggered.

### C. Method

The method FDM is named after three main steps: noise filtering, incident pattern discovery, and incident pattern matching. The first two steps are utilized to extract incident patterns during the offline phase and the last step is applied for the online detection phase. We depict the overview of FDM in Figure 1. We will explain the individual parts of Figure 1 throughout this section.

*1) Noise Filtering:* First, we extract incident speed vectors from trajectories that pass through incident locations when an incident occurred. This step is depicted in the upper right part of Figure 1. We normalize the incident speed and normal speed vectors by dividing each coefficient by the respective largest coefficent. After the normalization step, we filter out incident vectors which are too similar to their corresponding normal speed vector. This step is necessary as not all vehicles passing through an incident location are actually negatively affected by the incident (e.g., the incident might already be resolved by the time the vehicle passes by, or the incident was not severe enough to affect the traffic). For this, we compare the $L_1$ similarity against a given distance threshold $\delta$. Incident speed vectors with a $L_1$ smaller than $\delta$ are not included into the generation of the incident patterns.

*2) Incident Pattern Discovery:* To obtain the incident patterns $\overrightarrow{P_{a_1}}, \ldots, \overrightarrow{P_{a_k}}$ from the set of all candidate patterns $\overrightarrow{P_{cd}}$, we group them into $k \in \mathbb{N}$ clusters w.r.t $L_1$ similarity and choose the center of these $k$ clusters as the incident patterns. For clustering, we use K-means. The clustering process which results in the incident patterns is depicted in the upper left part of Figure 1.

*3) Pattern Matching:* After we have obtained the incident patterns from the offline discovery phase, we can use these patterns to classify new patterns observed in the online phase (lower part of Figure 1). Given a vehicle $T$, we calculate the *speed vector for* $T$ according to our road segments and the corresponding normal speed vector for $T$, based on the historical data for the same road segments. The difference between the speed vector for $T$ and the normal speed vector for $T$ gives us the *pattern for* $T$. We again use the same normalization technique as in Section III-C1.

Given the pattern for trajectory $T$, we measure the distance between it and the $k$ incident patterns using $L_1$ distance. If there exists one incident pattern such that its distance to the pattern for trajectory $T$ is smaller than a predefined threshold $\gamma$ (condition in Equation (1)), then we have detected an incident. As soon as we have found one pattern which satisfies the condition, we can stop the computation:

$$\exists\ i : d(\overrightarrow{P_T}, \overrightarrow{P_{a_i}}) < \gamma\ , i \in \{1, 2, ..., k\}\ , \tag{1}$$

where $d(\overrightarrow{P_T}, \overrightarrow{P_{a_i}}) = \sum_{j=1}^{m} |\overrightarrow{P_{T_j}} - \overrightarrow{P_{a_{ij}}}|$.

## IV. Experiments

### A. Dataset

We conduct experiments on two GPS datasets. The first data set (**HK**) consists of 35.1 gigabytes of trajectory data collected from 440 taxis in Hong Kong in the year of 2010. The generation rate of the GPS positions is around one position every 40 seconds. The city map of Hong Kong is collected from OpenStreetMap[4]. We also obtain incident data from the Transportation Department of Hong Kong for the year 2010. From this data, we use 4,386 incidents which occurred in road segments which are visited by at least one taxi during the specified time window.

The second dataset (**BJ**) simulates GPS data and incident data using the well-known simulation software Simulation of Urban MObility (SUMO) [12]. SUMO is used in various work on traffic analysis [8]. We follow the same setting as in *Traffic State Estimation* [8] to simulate GPS data and incident data in the city map within the second ring road in Beijing. We simulated 5,000 incident data and 4.41 gigabytes of GPS data.

In addition to the incident instances, we choose randomly non-incident instances such that both datasets are balanced with a 1:2.3 split [4], [13] between incident and non-incident instances.

We preprocess our datasets in four steps. We first partition the whole road network into road segments not longer than 100 meters. Then, incidents are matched to road segments. We filter out incidents whose distances to their closest road segments exceed 100 meters. Then, we conduct map matching by applying the algorithm in [14]. Third, we use linear interpolation to estimate the speed of a vehicle for segments in which we don't have a GPS signal. Fourth, we randomly split the data into 80% for the discovery phase (training) and 20% for the online detection phase (testing).

[4] https://www.openstreetmap.org

### *B. Competitors*

We compare our method with five advanced competitors that are designed for trajectory data or sensor data, which are SVMN [2], NNA [3], Topic Model (TM) [9], CNNU [11], Traffic State Estimation (TSE) [8].

### *C. Performance Metrics*

Following [2], [3], [8], [11], we evaluate our model based on detection rate (DR), false alarm rate (FAR), mean time-to-detect (MTTD), and F1 score:

$$\text{DR} = \frac{\text{number of detected incidents}}{\text{total number of tested incidents}} , \tag{2}$$

$$\text{FAR} = \frac{\text{number of false alarms}}{\text{total number of tested non-incidents}} , \tag{3}$$

$$\text{MTTD} = \frac{1}{n}\sum_{i=1}^{n}(t_i^{\text{detected}} - t_i^{\text{occurred}}) , \tag{4}$$

where $t_i^{\text{detected}}$ is the time when the incident is detected, $t_i^{\text{occurred}}$ is the time when the incident occurs, and $n$ is the number of correctly detected incidents.

### *D. Evaluation Results*

For evaluation on efficiency and effectiveness, we choose a time window of 5 minutes, the number of segments we observe $m = 15$, and $k = 100$ incident patterns as default settings. We will show the evaluation over different parameter settings. The evaluation is conducted on a machine with a 2.2 GHz Intel Core i7 CPU and 16 GB 2400 MHz DDR4 Memory.

We list the evaluation results in Table I and II by comparing our method (FDM) with the competitors in both the **HK** and **BJ** datasets. As one can see, our method has the highest F1 score. In addition, our method has the lowest FAR and MTTD. Finally, our method has the highest DR for the **BJ** dataset and is still on a competitive level on the **HK** dataset. We want to point out that a low FAR is especially important as, in real-world applications, most investigated trajectories are expected to be non-incidents, rather than incidents.

## V. Conclusions

In this paper, we study the problem of traffic incident detection on urban roads on a city-wide scale. We propose a new model (FDM) that is inspired by an insight from the transportation field. We give a formal definition of our setting and provide a detailed performance analysis of our model compared with other state-of-the-art methods. Moreover, we evaluate our method according to different parameter settings by extensive experiments. Our results show that our method is more effective while having the lowest MTTD among all competitors. Thus, our method is a feasible solution to incident detection in settings with sparse trajectories on a city-wide scale.

TABLE I
Comparison with Competitors in HK

| Methods | F1 | DR | FAR | MTTD (min) |
|---|---|---|---|---|
| TM [9] | 72.9 | 71.1 | 23.9 | 2.5 |
| TSE [8] | 62.0 | 51.7 | 15.0 | 2.6 |
| CNNU [11] | 70.8 | 79.7 | 42.7 | 5.0 |
| SVMN [2] | 70.9 | **85.2** | 55.2 | 5.0 |
| NNA [3] | 70.6 | 83.2 | 52.3 | 5.0 |
| **FDM** | **79.4** | 75.8 | **14.9** | **2.27** |

TABLE II
Comparison with Competitors in BJ

| Methods | F1 | DR | FAR | MTTD (min) |
|---|---|---|---|---|
| TM [9] | 74.4 | 71.7 | 20.9 | 2.4 |
| TSE [8] | 85.6 | 83.1 | 10.9 | 2.5 |
| CNNU [11] | 73.4 | 73.4 | 26.6 | 5.0 |
| SVMN [2] | 67.5 | 65.9 | 29.2 | 5.0 |
| NNA [3] | 63.0 | 60.7 | 31.8 | 5.0 |
| **FDM** | **89.9** | **86.4** | **8.69** | **1.1** |

## Acknowledgement

Reynold Cheng, Xiaolin Han, Tobias Grubenmann, and Xiaodong Li were supported by the Research Grants Council of Hong Kong (RGC Projects HKU 17229116, 106150091, and 17205115), the University of Hong Kong (Projects 104004572, 102009508, and 104004129), and the Innovation and Technology Commission of Hong Kong (ITF project MRP/029/18). We also thank Dr. Yiu and Dr. Lam et al. for their insightful discussions in the early phase of this work.

## References

[1] S. Ghosh-Dastidar and H. Adeli, "Wavelet-clustering-neural network model for freeway incident detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 18, pp. 325–338, 2003.

[2] T. Šingliar and M. Hauskrecht, "Learning to detect incidents from noisily labeled data," *Machine learning*, vol. 79, no. 3, pp. 335–354, 2010.

[3] D. Srinivasan, X. Jin, and R. L. Cheu, "Adaptive neural network models for automatic incident detection on freeways," *Neurocomputing*, vol. 64, pp. 473–496, 2005.

[4] F. Yuan and R. L. Cheu, "Incident detection using support vector machines," *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 3-4, pp. 309–328, 2003.

[5] W. Wong and S. C. Wong, "Evaluation of the impact of traffic incidents using gps data," *Proceedings of the Institution of Civil Engineers-Transport*, vol. 169, pp. 148–162, 2016.

[6] H. J. Payne and S. C. Tignor, "Freeway incident-detection algorithms based on decision trees with states," *Transportation Research Record*, no. 682, 1978.

[7] Y. J. Stephanedes and J. Hourdakis, "Transferability of freeway incident detection algorithms," *Transportation research record*, 1996.

[8] E. D'Andrea and F. Marcelloni, "Detection of traffic congestion and incidents from gps trace analysis," *ESWA*, vol. 73, pp. 43–56, 2017.

[9] A. Kinoshita, A. Takasu, and J. Adachi, "Real-time traffic incident detection using a probabilistic topic model," *IS*, vol. 54, no. 1, 2015.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *JMLR*, vol. 3, no. Jan, pp. 993–1022, 2003.

[11] L. Zhu, F. Guo, R. Krishnan, and J. W. Polak, "A deep learning approach for traffic incident detection in urban networks," in *ITSC*, 2018.

[12] P. A. Lopez, M. Behrisch, and B.-W. et al., "Microscopic traffic simulation using sumo," in *ITSC*. IEEE, 2018, pp. 2575–2582.

[13] J. Wang, X. Li, S. S. Liao, and Z. Hua, "A hybrid approach for automatic incident detection," *TITS*, vol. 14, no. 3, pp. 1176–1185, 2013.

[14] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *GIS*, 2009, pp. 336–343.