Crowd Sensing of Traffic Anomalies based on Human Mobility and Social Media

Bei Pan[1] * † Yu Zheng[2] David Wilkie[3] * Cyrus Shahabi[1] †

[1]University of Southern California, Los Angeles, USA
[2]Microsoft Research, Beijing, China
[3]University of North Carolina, Chapel Hill, USA
beipan@usc.edu, yuzheng@microsoft.com, wilkie@cs.unc.edu, shahabi@usc.edu

ABSTRACT

The advances in mobile computing and social networking services enable people to probe the dynamics of a city. In this paper, we address the problem of detecting and describing traffic anomalies using crowd sensing with two forms of data, human mobility and social media. Traffic anomalies are caused by accidents, control, protests, sport events, celebrations, disasters and other events. Unlike existing traffic-anomaly-detection methods, we identify anomalies according to drivers' routing behavior on an urban road network. Here, a detected anomaly is represented by a sub-graph of a road network where drivers' routing behaviors significantly differ from their original patterns. We then try to describe the detected anomaly by mining representative terms from the social media that people posted when the anomaly happened. The system for detecting such traffic anomalies can benefit both drivers and transportation authorities, e.g., by notifying drivers approaching an anomaly and suggesting alternative routes, as well as supporting traffic jam diagnosis and dispersal. We evaluate our system with a GPS trajectory dataset generated by over 30,000 taxicabs over a period of 3 months in Beijing, and a dataset of tweets collected from WeiBo, a Twitter-like social site in China. The results demonstrate the effectiveness and efficiency of our system.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-*data mining, Spatial database and GIS*.

General Terms

Algorithms

*The work was done when the first and third authors were interns in Microsoft Research Asia under the supervision of the second author.

†The authors' work has been funded in part by NSF grants IIS-1115153 and IIS-1320149.

SIGSPATIAL GIS '13, November 05-08, 2013. Orlando, FL, USA

http://dx.doi.org/10.1145/2525314.2525343 .

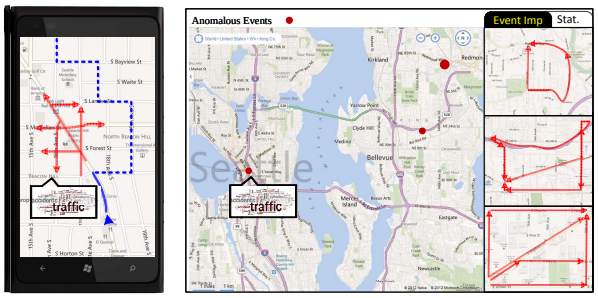
Keywords

city dynamics, human as a sensor, human mobility, traffic anomaly

1. INTRODUCTION

The prevalence of mobile phones, GPS, and social networking services has enabled people to probe the rhythm of the cities in which they live, becoming both smart sensors as well as actuators. The use of crowd sensing to capture the state of the city is a transforming paradigm, allowing real-time analysis and improved understanding and planning. One area of increasing interests is to understand the dynamics of urban traffic. Incidents and events can cause anomalies resulting in traffic jams on road networks that otherwise operate efficiently, costing time and money as well as increasing urban pollution. Additionally, the life of the city is often reflected in traffic patterns: popular sporting events draw crowds, holidays create disruptions, protests may result in road closures, etc. We propose to identify disruptions in the typical traffic patterns – traffic anomalies – and to give semantic meaning to the anomalies using two forms of crowd sensing, derived from GPS trajectories and social media contents.

In this paper, we propose a method to detect and describe traffic anomalies, which could be caused by traffic accidents, traffic controls, celebrations, protests, disasters, etc., by using human mobility data (such as GPS trajectories of vehicles) and social media data. Towards this end, we mine GPS trajectory data to detect significant routing changes; the subgraph of the road network on which an anomaly is found is then used to retrieve relevant social media to describe the anomaly. We envision two use cases for our system, one is oriented toward an individual user traveling around an anomaly and one is oriented toward city planners and traffic controllers to facilitate monitoring and visual analysis. Through the mobile user interface, our system provides services to individual users: 1) real-time alerts showing the anomaly area (i.e.



(a) Mobile UI (b) Web UI

Figure 1: System Prototype

represented by the red lines in Figure 1(a)) when users are nearby; 2) estimated features of the anomaly such as velocity and routing changes; and 3) a semantic context to give meaning to the anomaly, i.e. the social media terms that describe the event. Through a web-based user interface, our system provides the transportation authorities with a global view of all the traffic anomalies in the city, as shown in Figure 1(b). In this view, the anomalies are represented as red dots, and the detailed anomaly areas for each event are shown on the right column of the UI. Provided with such a service, transportation authorities could efficiently monitor all the traffic anomalies with detailed diagnoses of their impact regions and relevant descriptive terms.

Our system uses a novel methodology to detect anomalies according to drivers' routing behavior, i.e. the topological variation in traffic flow between points. This is different from related works on traffic anomaly detection, which focus on traffic volume and velocity on roads. Here, a detected anomaly is represented by a subgraph of a road network where people's routing behaviors significantly differ from their typical patterns. Figure 2 gives a concrete example, where 200 drivers travel from an origin *O* to a destination *D* in a period of day. As demonstrated in Figure 2 (a), normally, 80% of drivers go to *D* via route rt_1 while 10% travel along route rt_2 and 10% via route rt_3. Figure 2 (b) shows one kind of anomaly in which the traffic volume decreased on each route. Figure 2 (c) illustrates another kind of traffic anomaly where the total traffic flow is the same as before but the routing behavior of drivers along these routes has changed. Specifically, the percentage of drivers choosing rt_1 decreased from 80% to 25%, while the traffic on rt_2 and rt_3 increases from 10% to 30%, respectively. At the same time, a new route rt_4 has emerged, attracting 15% of drivers.

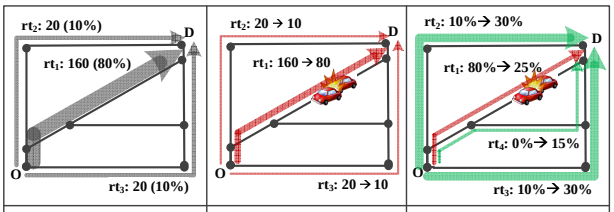


Figure 2: Concrete Example

Our approach has the following advantages over the existing methods. First, it provides a comprehensive view of the anomalies, showing the affected road segments of the anomaly as well as the relationships between these road segments. This is useful for diagnosing an anomaly or planning for traffic dispersal. For instance, traffic volume-based methods would only detect the road segment on which an accident has occurred, while other routes such as rt_2 and rt_3 would be overlooked. In fact, the traffic volume-based method may not even be able to detect some extreme cases, where the traffic volume does not change significantly on each road segment. Second, by detecting a subgraph, we enable the retrieval of relevant social media to describe the event. Without finding this geographic constraint and its time span, determining what social media is relevant to an anomaly would be far more costly, if not impossible. We use the historical tweets associated with the spatial region to represent the historical norm and report the terms that occur more frequently during the timespan of the anomaly as compared to their historical occurrences for this region.

The contributions of our paper are as follows:

- We present a novel method to detect traffic anomalies according to the routing behavior, showing significant advantages over traffic volume-based anomaly detection methods, e.g., revealing the affected spatial regions and relations between individual road segments, displaying potential alternative routes, and detecting anomalies that do not disrupt the traffic velocity or volume.
- We leverage social media to provide descriptions for the anomalies. By doing so, we correlate disruptions in the traffic patterns with their semantics, i.e. the urban events causing the anomalies.
- We validated our system using a large-scale, real-world GPS trajectory dataset generated by over 30,000 taxicabs over a period of 3 months in Beijing, which constitutes approximately 20% of the traffic flow [16] on the road network. Our results show that our method outperforms the baseline methods, including a traffic volume-based method.

The reminder of the paper is organized as follows. In Section 2, we overview the system and introduce the preliminaries. In Section 3, we explain our offline mining approach. In Section 4, we detail our anomaly detection approach. In Section 5, we present our system's capability to analyze the detected anomalies. In Section 6, we present the experimental setup and results. In Section 7, we summarize the related work. And in Section 8, we conclude our paper.

2. OVERVIEW

In this section, we define the terminologies used throughout the paper and give an overview of the system.

2.1 Preliminaries

Definition 1 (Road Segment): A road segment r is a directed edge in the road network graphs, with two terminal points $r.s$ and $r.e$. The vehicle flow on this edge is from $r.s$ to $r.e$.

Definition 2 (Road Network): A road network G is a directed graph, $G = (V, E)$, where V is a set of nodes representing the terminal points of road segments, and E is a set of edges denoting road segments.

Definition 3 (Path): A path p is a sequence of connected road segments, i.e., p: $r_1 \rightarrow r_2 \rightarrow ... \rightarrow r_n$, where $r_{(k+1)}.s = r_k.e$, $(1 \leq k < n)$.

Definition 4 (Trajectory): A trajectory tr is a sequence of GPS points created by a moving object. Each point consists of a longitude, latitude and a time stamp (t).

In this work, we map-matched these GPS points onto a path in the road network, thereby, each trajectory can be converted to a set of time-ordered road segments, i.e., $< t_1, r_1 > \rightarrow < t_2, r_2 > \rightarrow ... \rightarrow < t_n, r_n >$, where $r_{(k+1)}.s = r_k.e$, and t_k indicates the arrival time on the road segment r_k $(1 \leq k < n)$.

2.2 System Overview

Figure 3 shows the architecture of our system, which consists of three parts: offline mining, anomaly detection, and anomaly analysis.

Offline mining: As illustrated in the left column of Figure 3, this step consists of identifying the normal routing behavior of drivers which happens in general cases (detailed in Section 3.2). This step accumulates historical mobility data (e.g., GPS trajectories from vehicles) into a trajectory database and builds an index between road segments and the trajectories traversing them in order to enable online anomaly detection (refer to Section 3.3). This step also calculates the number and travel times of vehicles traversing each road segment over the course of a day.

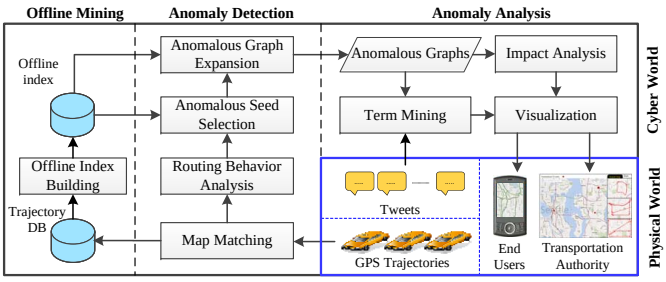


Figure 3: System Architecture

Online anomaly detection: As shown in the middle column of Figure 3, anomaly detection is an online inference step based on the recently received GPS trajectories of vehicles and the behavioral knowledge we obtained from offline mining. First, our system maps the received GPS trajectories of vehicles onto a road network using a map-matching algorithm presented in [13]. One copy of these processed trajectories is sent to the trajectory database for offline mining. Another copy is used for real-time routing behavior analysis. Similar to offline mining, we analyze the current vehicle flow and travel time for each road segment. By comparing the real-time information with our historical routing behavior knowledge, our system selects road segments with a certain deviation from its normal pattern (we call such such road segments *seed segments*). Then, our system expands each seed segment to a complete anomaly subgraph, over which drivers' routing behavior changed significantly (refer to Section 4 for details). Based on the offline index, an online indexing structure between paths and trajectories is built for efficient anomaly graph expansion.

Anomaly Analysis: On the right of Figure 3, the anomaly analysis step aims to analyze and explain the anomaly. One class of analytic information is the anomaly's impact in terms of the travel time delay on each path of the detected anomaly graph. We extract the information from the recent GPS trajectories of vehicles and the ordinary routing behaviors learned offline. Another class of analytic information is the representative terms (such as 'bridge out', 'accident', 'sports', etc.) that could describe or diagnose the anomaly. Specifically, we retrieve the relevant social media, e.g., tweets, using the time span when the anomaly occurred and the name of the roads covered by the anomaly. We then mine the representative terms that occurred frequently in the time span of the anomaly but rarely appeared otherwise. Finally, as a result, our system creates visualizations for individual drivers showing the extent of the anomaly as well as visualizations for more in-depth visual analysis.

3. OFFLINE MINING

3.1 Modeling Taxi Trajectories

We first partition the GPS logs from each taxi into independent trajectories representing individual trips, which is done using the taxi's transaction records. Next, we employ IVMM algorithm [13], to map each GPS point onto a road segment. Due to the fact that taxis normally report their GPS location every 1 to 2 minutes, these mapped road segments may not be connected with each other. Therefore, we connect each consecutive pair of GPS points with a path calculated based on the method described in [11]. As the result of this step, each trajectory has been converted to a directed path composed by connected road segments. For each trajectory, we also estimate the travel time on each road segment in its mapped path. We assume the travel time between two GPS points is uniformly distributed over the connecting path.

3.2 Modeling Routing Behavior

We model the routing behavior between two points as the distribution of traffic flow across different connecting paths. The preliminaries for this model are provided as follows:

Definition 5 (Original Edge and Destination Edge): For an edge r in a graph G, if there are no incoming edges connected with $r.s$, r is denoted as origin edge (r_O). Similarly, if there is no outgoing edges connected with $r.e$, r is denoted as destination edge (r_D).

Definition 6 (Routing Pattern) : For each pair of $< r_O, r_D >$ in road network graph G, at time t, its Routing Pattern(RP) is defined as $< f_1, p_1, f_2, p_2, ..., f_m, p_m >$, where f_i is the traffic volume (i.e., number of vehicles) on the i-th path from r_O to r_D, and p_i is the percentage of the total flow (i.e., the sum of f_i) between r_O and r_D using the i-th path. Note that this definition can implicitly reflect the high-level heading factor on the road network.

Consider the graph in Figure 2 as an example. Suppose the time stamps for the three figures are t_1, t_2 and t_3. The traverse flows and the routing behavior (i.e., routing pattern) for these three cases between O and D are shown in Table 1.

Time	Routing Pattern (RP)
t_1	<160, 0.8, 20, 0.1, 20, 0.1>
t_2	<80, 0.8, 10, 0.1, 10, 0.1>
t_3	<50, 0.25, 60, 0.3, 60, 0.3, 30, 0.15>

Table 1: Example of RP_{OD}

To measure the differences of routing behavior at time t_1 (RP_{t_1}) with another routing behavior (RP_{t_2}), we define the Mahalanobis distance [3] as follows:

$$d_M(RP_{t_1}, RP_{t_2}) = \sqrt{(RP_{t_1} - RP_{t_2})^T S^{-1} (RP_{t_1} - RP_{t_2})} \quad (1)$$

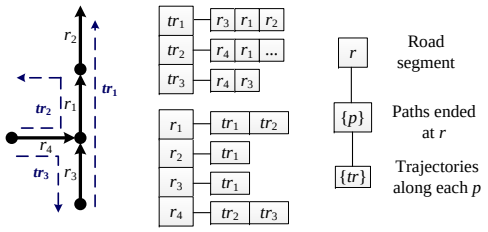
where S represents the covariance matrix between vector RP_{t_1} and RP_{t_2}. The reason why we choose Mahalanobis distance measurement lies in its capability in correlation analysis, through which different patterns in routing behavior can be identified. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant. Note that if the length of two routing patterns are different (e.g., the routing patterns for t_2 and t_3 in Table 1), additional zeros will be appended to the shorter vector to match the size of the longer one.

3.3 Index Building

We create two index structures, an offline index and online index, for speeding up the anomaly detection process.

The offline index is a bi-directional index structure between the trajectories and road segments. As stated in Section 3.1, each trajectory is converted into a path of connected road segments. The indexing in the *forward* direction is between each distinct trajectory and all the road segments contained in the derived path. In the *reverse* direction, each road segment is indexed by every trajectory that traversed it. Consider the example on Figure 4(a), where solid directed lines represent road segments, and the dash lines represent the trajectories. The corresponding offline index is depicted in Figure 4(b). This index structure is built offline, but will be updated online as new trajectories are received.

Our system also includes an online index. The online index is a index structure created for each road segment r to index all the ended paths and the trajectories along them. Note that *ended paths* refers to the paths which include r as the last edge. The structure of the online index is depicted in Figure 4(c). To detect an anomaly, we must efficiently find all the trajectories on the set of road segments and examine their routing behaviors. However, if



(a) Example **(b) Offline Index Eg.** **(c) Online Index**

Figure 4: Example of Index

we index the trajectories on all possible combinations of road segments within the road network, the size of the index entries would be exponential relative to the size of road network. Therefore, during the anomaly detection process, we use the offline index to build an additional index structure to expedite the searching process. The details of construction and maintenance strategy will be discussed in following section.

4. ANOMALY DETECTION

Problem Definition: Given a road network graph G composed of a set of road segments $R = \{r_1, r_2, ..., r_n\}$, a set of drivers $U = \{u_1, u_2, ..., u_m\}$, and a set of trajectories $TR = \{tr_1, tr_2, ..., tr_h\}$ during $[t_0, t_1]$, find a set of subgraphs, called traffic anomaly graphs, $\{g_1, g_2, ...g_k\}$, where each graph g_i satisfies the following criteria:
(a) g_i is a connected graph
(b) In g_i, for each r_O, there is at least one r_D, such that the routing pattern (*RP*) at t_1 between r_O and r_D, satisfies the following:

$$d_M(RP_{t_1}, \mu_{[t_0,t_1)}) \geq 3 \cdot \sqrt{\frac{1}{N} \sum_{t \in [t_0,t_1)} (RP_t - \mu_{[t_0,t_1)})^2} \quad (2)$$

where $\mu_{[t_0,t_1)}$ denotes the median of all routing patterns calculated during $[t_0, t_1)$. Similarly, for each r_D, there is at least one r_O satisfied the above constraint.

In (2), the left side is the Mahalanobis distance between the routing pattern at t_1 and the regular routing pattern[1]. The right side is three times of standard deviation from all routing patterns before t_1. Thereby, conceptually, we aim at detecting traffic anomalies with routing pattern highly deviated from regular routing pattern. Towards this end, we select seed segments and expand them into subgraphs that satisfy the conditions described above. In the following two sub-sections, we will discuss the two major steps, anomalous seed selection and graph expansion.

4.1 Anomalous Seed Selection

Definition 7 (Anomalous Seed Segment): At time *T*, an anomalous seed segment is a single road segment that is not in any existing anomaly subgraph and has a current flow (f_T) that satisfies the following equation:

$$|f_T - \mu| \geq 3 \cdot \sqrt{\frac{1}{N} \sum_{i=t_1}^{t_N} (f_{t_i} - \mu)^2} \quad (3)$$

where μ is the historical median flow on the road segments, and t_1 to t_N refer to all the time stamps before T.

At time *T*, all the road segments that satisfy the above definition are put into a pool of anomalous seed candidates. During the selection procedure, we sort the road segments in the pool such that they are ordered according to the scale of their flow variation, with the segment with the greatest variation being first in the queue. To ensure the selected seeds are not in any existing anomaly graphs, we select one seed at a time and expanding the corresponding anomaly graph, then iterate the process. Intuitively, for several road segments with similar flow changes located near to each other, it is entirely possible that their changes indicate they are part of the same anomaly. Thereby, instead of identifying multiple seeds at once, we select one seed and expand from it. After we finish expanding the anomaly subgraph, all of its road segments that are existed in the candidate pool are removed from the pool. The seed selection process terminates when the pool is empty. Note that, to measure the flow variation, we use relative variation rather than absolute variation, and in practice, we discard the road segments with extremely low historical flow from seed candidate pool as too little is known about them.

[1]To calculate the regular route pattern, we first derive the median traffic volume on each path, and then calculate the percentage based on the sum of median volume.

4.2 Anomalous Graph Expansion

With the selected anomalous seed segment, the next step is to expand from the seed to find the complete anomaly graph. The details of the expansion procedure are described in Algorithm 1. In general, it is based on breadth first graph expansion algorithm, with a verification step as pruning based on the criteria (b) in the problem definition.

Algorithm 1 MSGDetector($seed$, *R*, *Tr*, t, α, ϕ)

Output: Mobility Shift Graph: g

```
1: Let Graph g = InitGraph(seed)
2: Let Queue q = InitQueue();
3: q.enqueue(seed.neighbors(R));
4: while q ≠ ∅ do
5:    RoadSegment r = q.dequeue();
6:    g.addEdge(r);
7:    bool p = Verification(g, Tr,t, α, φ)
8:    if p == true then
9:       /// criteria (b) in problem definition is satisfied
10:      q.enqueue(r.neighbors(R));
11:   else
12:      g.deleteEdge(r);
13:   end if
14: end while
15: Return g
```

The verification step of Algorithm 1 is computationally costly, because it first needs to retrieve all paths between all O-D pair in the anomaly graphs (i.e., all combinations of r_O and r_D), and then record the number of traversing vehicles both historically and at present for each path to calculate the routing pattern. Specifically, the most computational costly step is to find all the historical trajectories traversing a path. Since a path is normally composed of several road segments, the offline index on individual road segment cannot help directly. Therefore, we use an online index structure (depicted in Figure 4(c)) to reduce the complexity of retrieving the trajectories, as described in the following paragraphs.

For each insertion of a new road segment, we need to update the relevant paths and trajectories in the online index structure. Consider a sample insertion sequence as depicted in step 1 to 4 in the Figure 5, the corresponding procedure of building the index is shown in Figure 6. The complexity of each insertion is shown in Table 2, where Tr_{xyz} indicates the trajectory group which is indexed by the path $r_x r_y r_z$. The function *F(A,r)* returns all the trajectories in set *A* that passed edge r, and *TR* refers to all historical trajectories. There are three possible positions for the newly added edge in a graph: as a destination edge(r_D), an origin edge(r_O), or

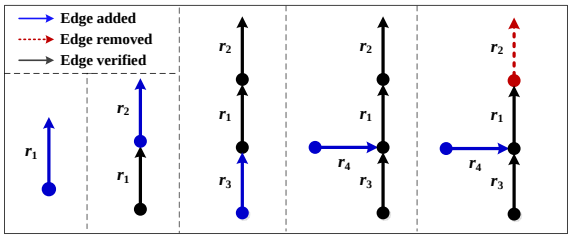


Figure 5: Sample Insertion Procedure

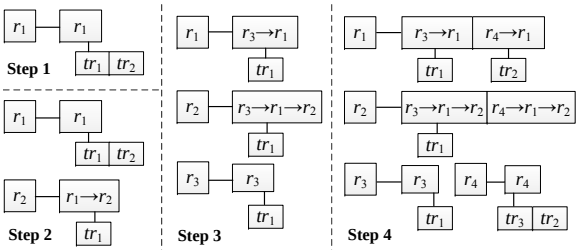


Figure 6: Update Procedure for Online Index

	Updated Index			Complx.
	Edge	Paths	Trajectories	
(1)	r_1	$+ r_1$	Tr_1 = F(TR, r_1)	$O(1)$
(2)	r_2	$+ r_1 r_2$	Tr_{12}=F(Tr_1, r_2)	$O(\|Tr_1\|)$
(3)	r_3	$+ r_3$	Tr_3= F(TR, r_3)	$O(1)$
	r_1	$- r_1$ $+ r_3 r_1$	Tr_{31} =F(Tr_1, r_3)	$O(\|Tr_1\|)$
	r_2	$- r_1 r_2$ $+ r_3 r_1 r_2$	Tr_{312} = F(Tr_{12}, r_3)	$O(\|Tr_{12}\|)$
(4)	r_4	$+r_4$	Tr_4 = F(TR, r_4)	$O(1)$
	r_1	$+ r_4 r_1$	Tr_{41}=F(Tr_4, r_1)	$O(\|Tr_4\|)$
	r_2	$+ r_4 r_1 r_2$	Tr_{412}=F(Tr_{41}, r_2)	$O(\|Tr_{41}\|)$

Table 2: Computational Analysis of Update Procedure

an ordinary edge. The complexity regarding the insertion of new r_D and r_O will be discussed in the following, while the complexity of inserting an ordinary edge can be derived by a combination of the other two.

Destination Edge Insertion: For the insertion of new r_D, we need to create its own index without updating any other edge in the graph. For example, in step 2 (for the insertion of r_2), first we need find its incoming edges (i.e., $\{r_1\}$) from the existing graph G. Then, we need to append the new edge to the end of the paths from r_1 and retrieve all trajectories that pass through the new edge. In this example, the complexity is $O(|Tr_1|)$.

Origin Edge Insertion: For the insertion of new r_O, we need to create r_O's index as well as update indexes for other relevant edges. Here, the relevant edges refer to all edges reachable from r_O. There are two types of update operations depending on whether the new edge replaced the previous origin edge or not. The steps 3 and 4 in the above example illustrate the two situations respectively.

For the insertion of r_3 in step 3, as shown in the Figure 5, it replaces the existing origin edge of r_1. In this case, we create an index for r_3 and update the existing indexes for r_1 and r_2. For r_3, we use the same strategy as in step 1. For r_1 and r_2, we need to insert r_3 before the existing paths in each of their indexes, and search within their indexes to find trajectories that traverse r_3. As the operations for the three edges, r_1, r_2 and r_3, are independent of each other, thereby the updates can be accomplished in parallel. Hence, in general, the complexity of insertion in such cases is the maximum number of trajectories stored in the index of reachable road segments. For example, in this example, the overall complexity is $O(\max(|Tr_1|, |Tr_{21}|)$.

For the insertion of r_4 in step 4, since it does not replace any existing origin edges, we do not need to update existing indexes. Instead, we only need to add more entries in the index structure for reachable edges, r_1 and r_2. As shown in Table 2, this step involves the same three operations as step 3. However, the three operations here cannot be executed in parallel because the operation in the next step depends on the result from the previous step. For example, for the second operation, generating the index entry of r_1 relies on Tr_4 as the input, which is the result of first operation. In this way, the updates need to be executed sequentially. Therefore, the complexity equals to the sum of updating costs from the insertion edge to all of its reachable edges. In this example, the overall complexity is the $O(|Tr_4| + |Tr_{41}|)$.

Origin Edge Insertion With Pruning: To reduce the number of sequential operations, we propose a pruning strategy based on the following intuition: if the routing behavior on a sub-path p does not present much variation, the routing behavior on the complete path containing p will not present much variation either. Thereby, instead of completing the updating operations, we can first test on the sub-path to see whether it satisfies the criteria for the verification, if not, we will prune it from the updating sequence. Consider the example depicted in Figure 5, if the path $r_4 r_1$ does not passed the verification step, the last operation in the Table 2 could be pruned. Meanwhile, the updates of index of r_2 could be pruned, as the red edge shown in the step 5 in Figure 5. As the result, the overall complexity for the insertion of r_4 could be reduced to $O(|Tr_4|)$.

5. ANOMALY ANALYSIS

5.1 Impact Analysis

We evaluate the impact of traffic anomalies in terms of the total travel time delay on the detected anomalous graph. The travel times for individual cars may have high variance, rather than staying around a static value, due to estimation error, different durations of traffic lights, driver preferences, etc. To address this, we defined the mean travel time (M) for a road segment over the time interval T as follows:

$$M(T) = \frac{\sum_{i \in T} f_i \cdot t_i}{\sum_{i \in T} f_i} \tag{4}$$

where f_i denotes the traffic flow along the road segment at time interval i in T, and t_i represents corresponding travel time for flow f_i. Using this, the travel time delay at time period T_1 compared with period T_2 for a road segment r can be calculated as below:

$$D_r(T_1, T_2) = \max\{0, M_r(T_1) - M_r(T_2)\} \tag{5}$$

To evaluate the total travel time delay for the traffic anomaly graph, we specify the T_1 in the above definition as the occurrence time period for the traffic anomaly and T_2 as the corresponding time period in the past. We then add up all the travel time delay (D_r) for each road segment in the traffic anomaly graph. For the examples in Figure 7, the sum of all the travel time delay on R_1, R_2 and R_3 is used as the impact parameter of the anomaly.

By using this measure, we can further evaluate the severity of the detected anomalies. For severe anomalies, not only does the routing behavior changes, but drivers encounter large travel time delays in the impact region. On the other hand, some anomalies exist only as routing behavior changes without severe delays. In general, we focus on severe anomalies as these incur a high cost to both the drivers and the city. Therefore, by using the travel time, we conduct a post selection step to filter out the non-severe anomalies.

Specifically, for a detected anomaly graph g, if the inequality (6) is not satisfied, we consider the anomaly as not severe.

$$D_g(T_1) \geq 3 \cdot std(\{t \in T_2 | M_g(t)\}) \quad (6)$$

where std refers to the standard deviation function, and the set passed to this function consists of the M values at different time intervals during historical period, T_2. After this filtering step, the selected anomalies not only present anomalous routing behavior, but can also be considered as an anomaly in terms of travel time. Note that, in the implementation of stand deviation function, we actually use the median of M values rather than the mean as the center.

5.2 Term Mining

The online social media (e.g., microblogging service) allow people to post information (tweets) reflect what they are looking, hearing, feeling. In other words, the people using such social media services can be regarded as a human sensor of physical world. This motivated us to retrieve the relevant information from the human sensors to describe the traffic anomaly.

Towards this goal, we utilize the location and time information obtained from the anomalous graph to eliminate the irrelevant posts, in order to enhance the searching efficiency. However, the remaining posts are still not necessarily relevant to the detected anomaly, because they may include some posts referring to other phenomenon which are commonly discussed all the time. For example, if an anomaly happens near a famous restaurant, during the occurrence time of the anomaly, not only will tweets discussing the traffic anomaly be posted, but also the tweets regarding the famous dishes in the restaurant. Thereby, to filter out the commonly discussed terms, we propose a strategy based on comparing the frequency of current tweets with historical tweets, to ensure the effectiveness of the retrieved information. We detail our strategy by utilizing the flow chart shown in Figure 7.

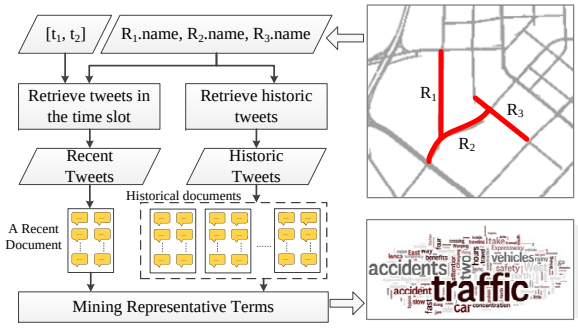


Figure 7: Term Mining Overview

Figure 7 shows the flow chart of the online tweets selection strategy for a sample traffic anomaly graph. As shown, the graph contains three edges, R_1, R_2 and R_3, and the corresponding time duration $[t_1, t_2]$. As illustrated in this figure, we first get the location information from the street names of each road segment in the anomalous graph. We use these names to collect all the tweets over a historical time interval (i.e., either the tweet is published at this location, or the content of the tweet refers to the street names in this location). We then use the time interval during which an anomaly was detected (i.e., $[t_1, t_2]$) to separate from the historical context the tweets that might relate to the anomaly. Here, we consider all the tweets posted in one day during $[t_1, t_2]$ as a document. For examples, the set of all the current tweets is considered as one document denoted as T_C. In this way, the historical tweets (T_H) refers to all the documents for each day in the past, as illustrated in Figure 7. Once we have both T_H and T_C, we analyze the relevance of each term among them using the strategy similar with *tf-idf* in [7]. Here, we have one document for the collection of current tweets (T_C). For historical tweets (T_H), we have a set of documents with each representing the collections of the tweets data during $[t_1, t_2]$ within one day in the past. Specifically, for each term, we calculate its relevance weights (w_t) as in Equation (7).

$$w_{term} = tf(term, T_C) \times idf(term, T_H)$$
$$s.t. \quad \begin{cases} tf(t,d) = \frac{f(t,d)}{\max\{f(w,d), \forall w \in d\}} \\ idf(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \end{cases} \quad (7)$$

where tf is the function to calculate the frequency of the term in the current tweet document (T_C), and idf refers to the calculation of inverse document frequencies in all the historical tweets documents (T_H). A high weight in Equation 7 is reached by a high term frequency (in the current tweets) and a low document frequency of the term in the whole collection of historical tweets. By using these weights, we can filter out the terms that frequently appear in the historical tweets. In the end, we ranked all the terms according to their weight to describe the anomalies. The term cloud in Figure 7, is one of our sample visualization based on the weights. The size of the terms is proportional to their weights.

To conclude, by identifying this geographic constraint and its time span, as well as guaranteeing the uniqueness of the terms, our approach is able to retrieve relevant social media (e.g., tweets) that offer description related to an anomaly. The efficiency and effectiveness of our approach are shown in the experiment section.

5.3 Visualization

Our system presents a visual representation of the discovered anomalies for users. We present a **navigation view** for use by drivers and a **analysis view** for planners. Our design is informed by work on stacked graphs [1], flow maps [10], and road network visualization [12].

Figure 8: Example of analysis view.

Our visualization shows a depiction of the road network overlaid on top of a satellite image. This serves to show the context of anomaly both in terms of the roads that are involved as well as the surrounding city geography. For the **navigation view**, seen in Figures 12(b), 13(b), we display an anomaly as a colored subgraph. The road segments of the anomaly are colored green, yellow, or red if the travel time is less than 2x, less than 3x, or greater than 3x the historical travel time, respectively. If travel time is not available, the segments are colored red if there is a decrease in flow and yellow otherwise. At each downstream boundary of the anomaly, the arrow represents the direction the traffic is flowing. For the **analysis view**, each road segment is additionally drawn with a width corresponding to the current flow and a width corresponding to the historical flow. The geometry representing the current flow is colored red, yellow or green, while the historical flow is colored black.

To demonstrate the **analysis view**, consider Figure 8. Here, we observe that the flow near the accident is less than the historical

flow, and the speed is over 3x slower. We can also see that the flow on the offramp has increased and is moving at at least half of the historical value. The flow has increased along some detour routes, where the speed has remained at least half of the historical. The speed on the onramp has dropped to less than a third of the historical, raising the possibility that the traffic jam could extend down the ramp and affect the crossing highway.

6. EXPERIMENTS

6.1 Dataset

Mobility Data: We use GPS trajectories as mobility data, with statistics shown in Table 3. As about 20% of traffic on road surfaces in Beijing is generated by taxicabs, the taxi trajectories represent a significant portion of the traffic flow on the road network. While we use taxi trajectory for validation, we believe our system and method are general enough to accept trajectory data generated by other sources, such as from public transit or location based check-in data, as long as they reflect mobility on the road network.

Road Network: We have the road networks of Beijing, with statistics shown in Table 3.

Traffic Anomaly Reports: We use the traffic anomaly reports published by transportation agencies as the ground truth to evaluate the effectiveness of our approach, the statistics is shown in Table 3.

	data duration	Mar-May, 2011
Trajectories	# of taxis	13,597
	# of effective days	51
	# of trips	19,455,948
	avg. sampling interval (s)	70.45
Roads	# of road segments	162,246
	# of road nodes	121,771
Reports	avg. # of reports per day	23

Table 3: Statistics of dataset

6.2 Evaluation Approach

In this study, we explore the effectiveness and efficiency of our approach to traffic anomaly detection as well as the efficiency of our approach to term mining to help analyze and describe the detected anomalies. In this experiment, we consider the traffic anomaly reported in last three weeks in the 3-month period as test data to evaluate the overall accuracy of our approach. In this evaluation, we study the performance of our method using a time discretization of 30 minutes. In other words, we carry out our method for anomaly detection every 30 minutes and consider the taxi trajectories collected during this time interval as current data, and all the trajectories collected before as historical data to calculate the regular routing behavior. According the study in [2], the length of a time interval is a trade-off between the computational load and the timeliness of an application.

Measurement: To evaluate the effectiveness of our approach, we consider the reported traffic incidents as a *subset* of ground truth, because the reported incidents are not necessarily a complete set of ground truth. We employ a parameter $recall$ to measure the accuracy of the detected anomalies. In our experiments, $recall$ is the fraction of the number of detected reported anomalies over the number of all the anomalies reported. Note that, in this evaluation, we did not use the *precision* measurement, since we consider the reported incidents as a subset of ground truth. It is entirely possible that some traffic anomaly, which resulted in the change of routing behavior and travel time delay, is detected by our approach but not reported by transportation authorities, such as the second case study presented in the result section.

Baselines: To evaluate the accuracy of our approach, we use a modified version of Principle Component Analysis (PCA) applied in [2] as a baseline anomaly detection approach. Unlike our work, this method focuses solely on traffic flow. The details of the implementation are as follows: we first applied PCA on a matrix of all road segments to find the anomalous road segments during a specific time period; then, we aggregate the nearby road segments into a connected graph as the anomalous graph. For the anomaly analysis, we consider an anomaly detection algorithm purely using social media similar with [8] as our baseline approach, which was initially proposed to detect the location and the description of earthquakes in real-time. This baseline approach uses keywords such as "earthquake" to filter the irrelevant tweets. However, in our case, there is no indication of what terms might be relevant to the anomaly. Therefore we cannot use pre-defined keywords to do the filtering. As a result, we use this approach without keyword-filtering step as our baseline.

6.3 Results

6.3.1 Effectiveness

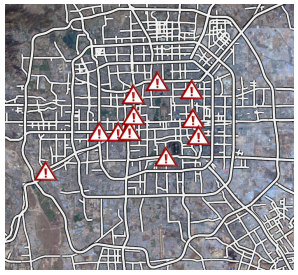
To evaluate our approach, we show the result under two 'rush hour' time intervals (i.e., 7-9AM and 4-6PM) on 5/12/2011 in Figure 9, where the caution label indicates the location of the anomalies. Figure 9 (a) and (b) show all the *reported* anomalies during the two time intervals; (c) and (d) show the anomalies detected by the *baseline* approach; and (e) and (f) show the detected anomaly by *our approach*.

As shown from Figure 9, in both time intervals, our approach detects more anomalies than the baseline approach. In particular, for 7-9AM interval, our approaches detected all the reported anomalies, but baseline only detects two of them. For 4-6PM, our approach detect 8 reported anomalies but baseline only detects 7 of them. Specifically, in this particular experiments, the recall value for our approach improves the baseline by 85.6%. In the evaluation of over all the test data (i.e., all the anomalies occurred in the last three weeks of the dataset), average recall value for our approach is 86.7%, while that for baseline is around 46.7%. Therefore, we claim our approach significantly outperforms the traffic-volume approach. We believe this is due to the fact that our approach can detect the anomalies reflected not only from the traffic volume change, but also from the change of routing behavior.

To further show the superiority of our approach, we choose a particular anomaly detected using our approach during 8:30AM to 9:00AM, but NOT detected by baseline approach in this time interval. In this case, based on our detected graph, there is a significant routing behavior shift from the main road (denoted as *M-routing*) to the auxiliary road (denoted as *A-routing*). In Figure 10, we visualize the change of overall flow (the sum of the flow on the two routes) as well as the change of the routing behavior between the main road and the auxiliary road over time.

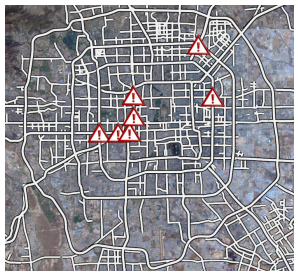
According to the Figure 10(a), during the interval of 8:30AM to 9:00AM, the overall flow bypassing the two routes did not show much difference compared with regular flow. However, during 9:00AM to 9:30AM, as people started to avoid the anomaly region, the overall flow decreases. On the other hand, the routing percentage changes in a different manner compared with that of overall flow. According to Figure10(b), people start to change their routes immediately after the anomaly happens (i.e., in the interval between 8:30AM to 9:00AM). During 9:00AM to 9:30AM, the routing behavior starts to recover to normal, while the overall flow in this region starts to behave abnormally. Therefore, in this case, when the sliding window reaches the time interval 8:30AM to 9:00AM,

(a) 7-9AM: Reported incidents



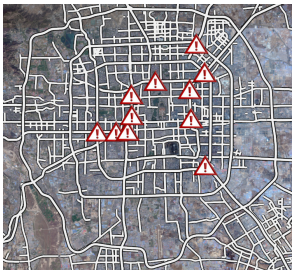
(b) 4-6PM: Reported incidents

(c) 7-9AM: Baseline results



(d) 4-6PM: Baseline results

(e) 7-9AM: Our results



(f) 4-6PM: Our results

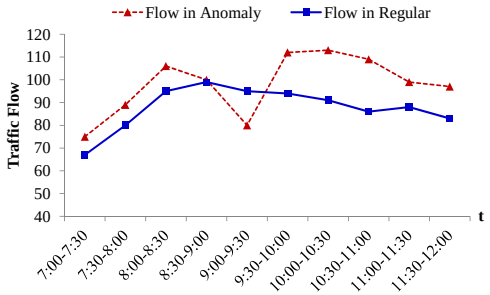
Figure 9: Traffic anomalies reported to authorities, discovered by the baseline PCA approach, and discovered by our method from 7AM to 9AM and from 4PM to 6PM on 5/12/2011

the baseline approaches cannot detect the anomaly as the overall traffic volume has not changed significantly. However, as our approach considers the change of routing behavior, it can identify the anomaly in a more timely fashion than baseline approach.

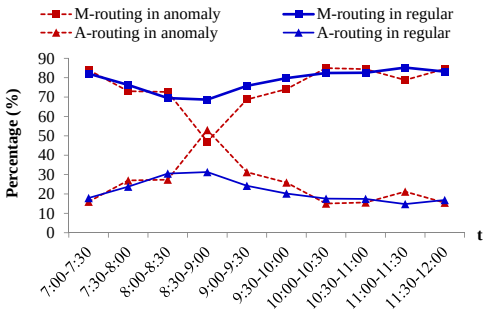
6.3.2 *Efficiency*

In this set of experiments, we compare our anomaly detection approach with the approach without online index structure. Both approaches are implemented on a 64-bit server running Windows Server 2008 (OS) using a single thread of a 2.66GHZ CPU with 16G memory. Figure 11 shows comparison result. As the size of detected mobility graph grows, our approach performs increasingly better than the approach without an index, due to the fact that no-index approach spends a great amount of time in verifications for all the O-D pairs during the expansion. However, our approach uses the additional data structure to avoid scanning every path between all the O-D pairs.

Table 4 shows the average processing times for major steps in anomaly detection. The map matching procedure is always running in the background as a pre-processor to convert each GPS trajectory we collect online. The average processing time for map-matching one trajectory is 0.085 seconds. Assuming the number of anomalies is less than 10 per 30 minute period, our system can detect these anomalies within 1 to 2 minutes. The efficiency of anomaly analysis based on social media is evaluated through the case studies.



(a) Routing Flow Comparison



(b) Routing Behavior Comparison

Figure 10: Effects of time intervals

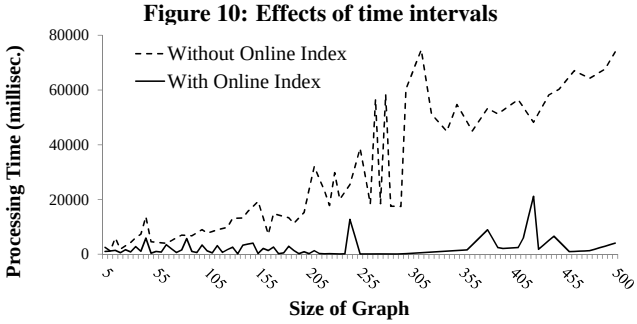


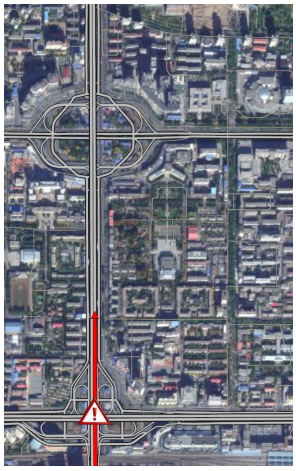
Figure 11: Effects of Index

Procedure Name	Time (s)
Routing Behavior Analysis	1.2
Anomalous Seed Selection	<1
Anomalous Graph Expansion	5.2

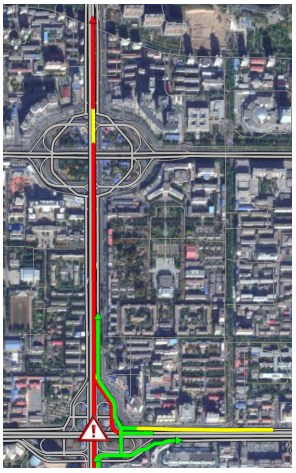
Table 4: Average processing time in anomaly detection

6.3.3 *Case Studies*

We further evaluate our approach using two case studies: one is reported and detected by our system, another one is detected by our system but is not reported. The reported anomaly is caused by a traffic accident, and the un-reported anomaly is probably caused by the wedding expo exhibition according to our analysis. The results for the these case studies are depicted in Figure 12 and 13, respectively. In these figures, (a) presents the detected anomalous graphs by the baseline approach and (b) presents the anomaly detected by our approach. On the anomalous graph, the red, yellow and green lines indicate the travel time metric, as described in Section 5.3, and the caution mark represents the location of the anomaly reported by the transportation authorities. In addition to the detected anomalous graphs, we also present the results from term analysis in the sub-figure (c). We also compare some relevant results during ordinary times versus during anomaly time in sub-table (d), such as the number of tweets and the *tf-idf* value of important terms.



(a) Detected by baseline



(b) Detected by our method

(c) Terms discovered

	ordinary	anomaly
#. of tweets	$1.6*10^2$	$1.9*10^2$
$w_{traffic}$	0.361	0.903
$w_{accident}$	0.202	0.978

(d) Relevant results

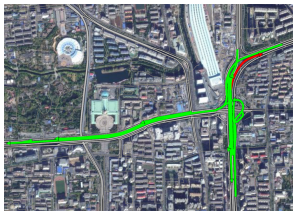
Figure 12: Case study 1 – Anomaly detected

For the first case, according to the anomaly report, during 4PM-4:30PM on 5/19/2011, a two-car accident occurred on the Lianhua bridge in the north-bound direction. In the anomalous graph (i.e., Figure 12 (a)) detected by the baseline approach, only a small part of the highway around Lianhua bridge is included. On the other hand, in our detected graph (i.e., Figure 12 (b)), a more comprehensive view is provided in the following two aspects: 1) we detect a larger and more complete region impacted by this anomaly 2) by showing the yellow and green road segments, we can provide end-users the routes (i.e., auxiliary lanes) to detour or avoid the regions covered by red lines. These routing suggestions are implied in our detected graph as many people change their routing behavior to avoid or escape the anomaly.

In addition, we show the corresponding top 50 terms mined for this cases in Figure 12(c). According to this figure, the most highlighted words are "traffic" and "accident", which is also consistent with the anomaly reports from Beijing Transportation Bureau. In addition, the result also reveals some other information relevant to the anomaly. For example, the terms "two", "vehicles", "car", which may indicate this accident is involved with two cars, also shows the consistency with the anomaly reports. Also, the term "north" indicates the direction of the lane where accident happens, as well as "rainy" reveals the weather information at that time when the anomaly happens. Figure 12(d) shows in the anomaly time, there is no significant increase of tweets referring the anomaly location, compare with that in ordinary times. However, the $tf\text{-}idf$ value of some terms changed significantly, such as "accident" and "traffic". By using the idea of $tf\text{-}idf$, our method can successfully identify the relevant terms.

In the second case, our approach detected an anomaly at 8:30AM to 9AM on 5/27/2011 near the location of Beijing Exhibition Center. There is no anomaly reports from transportation agencies at this particular time and location, however, based on our analysis through online social media, the 18th Beijing Wedding Expo is opened at 9AM at Beijing Exhibition Center. According to the local news, each year, the wedding expo attracts a lot of wedding

(a) Detected by baseline



(b) Detected by our method

(c) Terms discovered

	ordinary	anomaly
#. of tweets	$2.0*10^2$	$5.6*10^2$
$w_{wedding}$	0.121	0.447
w_{expo}	0.060	0.298

(d) Relevant results

Figure 13: Case study 2 – Anomaly detected

	Baseline		Our Approach	
	$\|T_H\|$	$\|T_C\|$	$\|T_H\|$	$\|T_C\|$
Case 1	9.1×10^7	1.7×10^6	9.7×10^3	1.9×10^2
Case 2	8.5×10^7	1.9×10^6	3.4×10^4	5.6×10^2

Table 5: Comparison based on #. of Tweets Used

related companies to exhibit and sell their products as well as thousands of young people as customers, which can be considered as a significant shopping event. As shown in Figure 13, our detected anomalous graph is also more comprehensive and informative than the graph detected by baseline approach.

Different from previous case, there is no official transportation reports for this detected anomaly. To understand the anomaly, we further conduct the terms analysis from the online social media as result shown in Figure 13(c). From this figure, the most frequent mined terms are "wedding" and "expo", which implies the cause of the detected anomaly. Also, the detected terms "promotions" and "shopping", may suggest this event have great deals that may attract a lot people to shop there. From these mined terms, we could inference the traffic anomaly (i.e., significant travel time delay) is caused by too many people attending the wedding expo at its opening time at Beijing Exhibition Center. To conclude, our system can not only detect the traffic anomalies reported by the transportation agencies, but also, which is more important, detect the anomalies that are not reported.

Table 5 shows a comparison of our approach with the baseline based on the number of tweets used regarding the three cases studied above. Here, $|T_H|$ denotes the number of tweets published historically at the time of the anomaly. For example, for the first case, T_H represents all the tweets posted during 4:30PM to 5PM at each day before 05/19/2011 in the historical dataset. $|T_C|$ denotes the number of tweets published at the time of the anomaly. As presented, for both two cases, the number of tweets we analyzed in our approach is significantly reduced from that of the baseline (e.g., from the level of 10^6 to as low as 10^2). Since our approach focused on the tweets that were relevant (i.e., both spatially and temporally) to the detected anomaly graph, the search space of tweets is largely reduced compared with the baseline approach.

7. RELATED WORK

7.1 Anomaly Detection using Traffic Data

The previous work on detecting anomalies using GPS data can be divided into two categories: 1) the studies on trajectory anomalies (e.g., [4, 14, 15]), and 2) the studies on traffic anomalies (e.g.,

our work and [2]). The works in the first category sought to find a small percentage of drivers whose driving trajectories is different compared with the broader population, which could result from fraudulent taxi driving behavior or some other anomalous cause. Our work belongs to the second category, and differs from the above methods in the following aspects. First, we aim to detect a *large* amount of drivers whose behavior is anomalous. Second, for anomalous trajectory detection, the comparison between the trajectories always happens between a small set of trajectories and the remaining trajectories at the same time and location. For our work, the traffic anomaly detection, the comparison happens between the current behavior of drivers and the historical driving behavior.

The most relevant works to our study, in terms of both data types and the definition of an "anomaly", are those focusing on traffic anomaly detection using GPS data (e.g., [6], [2]). Among these works, our paper can be distinguished in two ways. First, our approach is the first considering the change of routing behavior in addition to the change in traffic volume. Therefore, we have found that our approach has a higher detection rate as compared with an approach that only uses traffic volume changes. Further, our technique can provide users with detour routes to avoid or escape the congestion caused by a traffic anomaly, while the volume based approaches can only detect the locations of the anomalies, without revealing the whole extent throughout the road network. These two advantages were evaluated in experiment section. Finally, the granularity of our detected traffic anomaly is on the level of road segments instead of spatial regions. For example, the anomalous scenarios studied in [2] are inter-regions, making its results limited to very large scale events, such as *marathon race*, instead of road-segment-level traffic anomalies, such as traffic accidents.

7.2 Anomaly Detection using Social Media

Another line of related work is anomaly detection via mining social media content. Recently, microblogging services (e.g., twitter) have received much research attention in the fields of anomaly detection. Researchers consider the twitter posts (i.e., tweets) as real-time social streams and focus on analyzing the features of keywords in the specific context to detect events [8, 9, 5]. The key challenges in these works is to filter out the irrelevant contents in the tweets, which requires computationally expensive filtering, such as the Kalman filtering based model proposed by [8] and the Gibbs Random Field defined probabilistic model in [5]. However, in our work, by using the data collected from anomaly detection in addition to the social texts, we can narrow down the search space to a specific time and location, tremendously reducing the search space as compared with the traditional methods. We therefore only need to conduct a simple filtering technique to separate out the irrelevant contents, as discussed in anomaly analysis section.

8. CONCLUSION

We presented an approach that uses two forms of crowd sensing, combining mobility data with social media, to understand one aspect of urban dynamics. Specifically, we detected and described traffic anomalies using a novel approach based on the routing behavior of drivers. Our approach enabled us to discover an entire sub-graph of the road network associated with an anomaly. Subsequently, we proposed an approach to mine social media for terms that are constrained to the sub-graph geographically and temporally and correlated with the anomaly. We evaluated our system with a GPS trajectory dataset generated by over 30,000 taxis over 3 months in Beijing. We examined the effectiveness and efficiency of our system and compared our approach with a baseline method using traffic volumes. We observed that our system can detect more traffic anomalies than those of the baseline, identifying 86.7% of the incidents reported to the transportation authority as compared to the baseline's detection of 46.7%. Unlike existing work, our work can be utilized to provide individual drivers approaching an anomaly with a display of the extent of the anomaly as well as a timely alert. Our system can aid transportation authorities with anomaly diagnosis and dispersal by providing the impact area of an anomaly, the correlations between the traffic flow changes on different roads, and descriptions of possible causes of the anomaly. Our system can process an anomaly in approximately 6 seconds using a single core, allowing for effective, real-time detection. Fusing social media with mobility data allows us to observe one aspect of urban life in a finer granularity than previously possible, revealing the geographic extents, dynamics, and semantic of traffic anomalies. We plan to investigate the correctness and usefulness of the social media for explaining traffic anomalies in the future.

9. REFERENCES

[1] L. Byron and M. Wattenberg. Stacked graphs–geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 2008.

[2] S. Chawla, Y. Zheng, and J. Hu. Inferring the root cause in road traffic anomalies. In *ICDM '12*.

[3] R. De Maesschalck, D. Jouan-Rimbaud, and Massart. The mahalanobis distance. In *Chemometrics and Intelligent Laboratory Systems 50*, pages 1–18, 2000.

[4] Y. Ge, H. Xiong, C. Liu, and Z.-H. Zhou. A taxi driving fraud detection system. In *ICDM '11*.

[5] C. X. Lin, B. Zhao, Q. Mei, and J. Han. PET: a statistical model for popular events tracking in social communities. In *KDD '10*.

[6] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing. Discovering spatio-temporal causal interactions in traffic data streams. In *KDD '11*.

[7] J. Ramos. Using TF-IDF to Determine Word Relevance in Document Queries. Technical report, Department of Computer Science, Rutgers University, 2003.

[8] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW '10*.

[9] H. Sayyadi, M. Hurst, and A. Maykov. Event detection and tracking in social streams. In *ICWSM '09). AAAI*, 2009.

[10] K. Verbeek, K. Buchin, and B. Speckmann. Flow map layout via spiral trees. *IEEE Transactions on Visualization and Computer Graphics*, 2011.

[11] L.-Y. Wei, Y. Zheng, and W.-C. Peng. Constructing popular routes from uncertain trajectories. In *KDD '12*.

[12] D. Wilkie, J. Sewall, and M. Lin. Transforming gis data into functional road models for large-scale traffic simulation. *IEEE Transactions on Visualization and Computer Graphics*, 2012.

[13] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G.-Z. Sun. An interactive-voting based map matching algorithm. In *MDM '10*.

[14] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li. iBAT: detecting anomalous taxi trajectories from GPS traces. In *UbiComp '11*.

[15] J. Zhang. Smarter outlier detection and deeper understanding of large-scale taxi trip records: a case study of NYC. In *SIGKDD '12 Workshop on Urban Computing*.

[16] Y. Zheng, Y. Liu, J. Yuan, and X. Xie. Urban computing with taxicabs. In *UbiComp '11*.