Discovering Pollution Sources and Propagation Patterns in Urban Area

Xiucheng Li*
Nanyang Technological University
xli055@e.ntu.edu.sg

Yun Cheng*
Air Scientific
chengyun.hit@gmail.com

Gao Cong
Nanyang Technological University
gaocong@ntu.edu.sg

Lisi Chen
Hong Kong Baptist University
chenlisi@comp.hkbu.edu.hk

ABSTRACT

Air quality is one of the most important environmental concerns in the world, and it has deteriorated substantially over the past years in many countries. For example, Chinese Academy of Social Sciences reports that the problem of haze and fog in China is hitting a record level, and China is currently suffering from the worst air pollution. Among the various causal factors of air quality, particulate matter with a diameter of 2.5 micrometers or less (i.e., $PM_{2.5}$) is a very important factor; governments and people are increasingly concerned with the concentration of $PM_{2.5}$. In many cities, stations for monitoring $PM_{2.5}$ concentration have been built by governments or companies to monitor urban air quality. Apart from monitoring, there is a rising demand for finding pollution sources of $PM_{2.5}$ and discovering the transmission of $PM_{2.5}$ based on the data from $PM_{2.5}$ monitoring stations.

However, to the best of our knowledge, none of previous work proposes a solution to the problem of detecting pollution sources and mining pollution propagation patterns from such monitoring data. In this work, we propose the first solution for the problem, which comprises two steps. The first step is to extract the uptrend intervals and calculate the causal strengths among spatially distributed sensors; The second step is to construct causality graphs and perform frequent subgraphs mining on these causality graphs to find pollution sources and propagation patterns. We use real-life monitoring data collected by a company in our experiments. Our experimental results demonstrate significant findings regarding pollution sources and pollutant propagations in Beijing, which will be useful for governments to make policy and govern pollution sources.

CCS CONCEPTS

•**Applied computing** →*Environmental sciences;*

KEYWORDS

Pollution Sources; Propagation Patterns; Real Deployed System

ACM Reference format:
Xiucheng Li*, Yun Cheng*, Gao Cong, and Lisi Chen. 2017. Discovering Pollution Sources and Propagation Patterns in Urban Area. In *Proceedings of KDD ’17, August 13-17, 2017, Halifax, NS, Canada,* , 10 pages.
DOI: 10.1145/3097983.3098090

1 INTRODUCTION

Air quality, which is one of the most important environmental concerns in our planet, has deteriorated substantially over the past years in many countries. The issue is especially significant for developing countries. For example, Chinese Academy of Social Sciences reports that the problem of haze and fog in China is hitting a record level, and China is currently suffering from the worst air pollution since 1961 [16].

Among the various causal factors of air quality, particulate matter with a diameter of 2.5 micrometers or less (i.e., $PM_{2.5}$), has gained much attention recently. Because $PM_{2.5}$ is the tiny particle that can be easily absorbed by the human lungs, it has significant impact on our respiratory systems and gives rise to asthma, cardiovascular disease, blood diseases, lung cancer, etc. [21]

A recent research finds that the average life expectancy of residents in northern China would be 5.5 years shorter than that in southern China due to the higher prevalence of $PM_{2.5}$ linked diseases in northern China [1]. The problem of $PM_{2.5}$ is even more serious in metropolitan cities. For example, 51.8% of days in 2013 are stuck at “unhealthy” levels of $PM_{2.5}$ or worse in Beijing. Very often, residents in Beijing wear masks and social network websites are exploded with complaints about the heavy blanket of smog induced by the high concentrations of $PM_{2.5}$ [21].

As a result, people are increasingly concerned with the concentration of $PM_{2.5}$. Many cities have built stations for monitoring $PM_{2.5}$ concentration.Apart from monitoring, there is a rising demand for finding pollution sources of $PM_{2.5}$ and discovering the transmit of $PM_{2.5}$ based on the data of $PM_{2.5}$ monitoring stations. Specifically, knowing the pollution sources and transmission patterns can help governments take timely actions and make appropriate policies to reduce the level of $PM_{2.5}$. However, finding the sources of $PM_{2.5}$ and discovering the transmission patterns of $PM_{2.5}$ are very challenging due to the following reasons.

Dynamic Pollution Sources: Air pollution sources may change over location and time significantly. For example, the location and time of pollutant from vehicles are greatly influenced by the traffic conditions and rush hours, respectively. As another example, a steel factory will be considered as a pollution source only when

∗ equal contribution.

KDD ’17, August 13-17, 2017, Halifax, NS, Canada
© 2017 ACM. 978-1-4503-4887-4/17/08…$15.00
DOI: 10.1145/3097983.3098090

it emits waste gas, and factories may choose to emit at late night. Furthermore, many other random activities may happen and they can be pollution sources, such as crop stubble burning, fireworks, roadside barbecues, and construction dust [16].

Unpredictable Transmit: The spread of $PM_{2.5}$ is affected by various complex factors, including surrounding environments, the direction and speed of airflow, precipitation, air pressure, humidity, intensity of sunlight, etc. [12]. Existing meteorological techniques are not able to tell us exactly when a particular weather event will occur and how long it will last. Consequently, it is extremely hard to figure out the propagation paths of a pollution source.

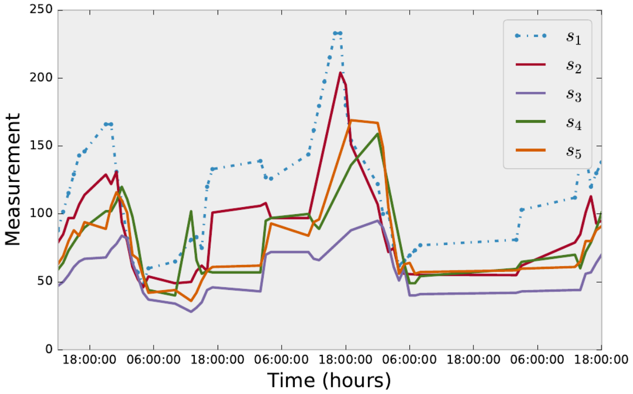


Figure 1: The $PM_{2.5}$ measurement of a group of sensors.

Recently, $PM_{2.5}$ monitoring sensors have been densely deployed in many cities, such as Beijing, to monitor the $PM_{2.5}$ concentration in fine-granularity [3]. Figure 1 depicts the $PM_{2.5}$ measurement of group of sensors, we can clearly see that the evolving behaviors (i.e., up and down behaviors) of s_1 usually preceded those of the other sensors. In particular, the increasing behaviors typically accompany with pollutant dispersion. Consequently, based on the observations and the measurement data in sensors, we propose to mine pollutant propagation patterns from the historical data of these sensors. With the mined propagation patterns, we are able to detect the potential pollution sources and their corresponding propagation paths.

There are two main technical issues for mining pollutant propagation patterns. The first is how to model "pollutant" and " pollutant propagation", respectively. As mentioned earlier, the source of pollutant may change significantly and the propagation of pollutant is affected by various factors. To address the challenge, we introduce the concept of *uptrend event* and use uptrend events to model pollutants. Next, we propose the measurement of *causal strength*, based on which we develop *spatio-temporal causality model* to represent the propagation between two uptrend events. Finally, we define the *casuality graph* to illustrate the potential propagations among each pair of uptrend events. The second issue is how to efficiently mine propagation patterns from the casuality graphs. We prove that all the constructed causality graphs are directed acyclic graphs. Based on the special property, we tailor an existing frequent subgraph mining algorithm to mine propagation patterns more efficiently.

To sum up, our contributions are two-fold:

(1) We study the new problem of mining pollutant propagation patterns and detecting pollutant sources based on the data from $PM_{2.5}$ and PM_{10} monitoring sensors. The proposed technique has been deployed by several local Environmental Protection Agency (EPA) in Beijing and Hebei, China, to help them make more intelligent decisions.
(2) In our experiments, we use two air quality monitoring datasets collected from 204 and 157 densely deployed monitors in Haidian district, Beijing, and the whole Beijing city, respectively. To evaluate the efficacy and efficiency of our mined propagation patterns we utilize them in temporal prediction task, which shows a big improvement compared with the baseline methods without using our discovered patterns. Our experimental results also demonstrate meaningful findings regarding the pollution sources and pollution propagations in Beijing.

2 RELATED WORK

Recently data centric approaches have been exploited to address environment related problems. Li et al. [9] estimated the air quality by analyzing the user-generated images. Shang et al. [15] attempted to infer the gas assumption and pollution emission of vehicles based on existing air pollution monitors. Cheng et al. [3] designed a cloud-based air quality monitoring system in which they fused the data from public monitors and a large amount densely deployed cheap sensors by using Gaussian Process to incorporate sensors with different confidence, Cheng et al. [2] also seeked to discover the dynamic co-evolving zones using the data collected by the deployed system. Several subsequent studies [7, 13, 14] then concentrated on improving the performance and function of this crowd sensing monitoring system to support more insightful analysis. However, to the best of our knowledge, our work is the first attempt to discover the potential pollution sources using the monitoring data from spatially distributed monitoring sensors within the urban area.

Our work is related to the work of Zhang et al. [20], which addresses a different problem. Specifically, they proposed an Assembler to uncover the co-evolving behaviors among massive geo-sensors, and the Assembler requires the co-evolving sensors to have common timestamps. Zhu et al. [23] studied the causalities of different pollutants with the Bayesian Network but they did not attempt to discover the pollution sources. Our approach comprises two main components. The first part involves the causality analysis in time series data. Granger Causality [6] was originally introduced to handle the time series analysis in econometrics. The assumption of Granger Causality is simple and has proven useful in many fields. Granger Causality was combined with graphical models by Lozano et al. [11] to model climate attribution and account for climate changes, where the authors assumed the same coefficient applies to each grid. In our approach, we follow the similar idea to model the causal strengths among geo-sensors. However, we relax the assumption and allow different geo-sensors to have their unique coefficients. The calculation of causal strengths can also be attributed to Elastic Net problem. Zou et al. [24] developed Elastic Net for the purpose of variable selection, and Elastic Net has then been widely used in machine learning and many well-developed packages emerged in the past years. In particular, a reduction has been made from Elastic Net to Support Vector Machines [22] which would enable many GPU accelerated tools designed for Support Vector Machines to tremendously speed up the computation of Elastic Net. To detect the spatio-temporal causal interactions in trajectory, Liu et al. [10] constructed a collection of outlier trees and mined the frequent substructures in the forest. However, unlike the trajectory of vehicles, the dispersion of pollutant does not have fixed paths and one location might be polluted by multiple sources and thus the tree structure is not suitable in our scenario. The second component of our method is related with the subgraph mining problem. There exists a host of work on mining different types of frequent subgraphs, such as the work [17–19, 25].

3 PROPOSED PARADIGM

Figure 2 demonstrates the framework of our model. The first step consists of two sub-steps: Extracting the uptrend intervals and calculating the causal strengths among spatially distributed sensors from the collected historical data. The geo-sensory time series data is overwhelmed by various small fluctuations. Therefore, we first remove the small fluctuations and extract the real evolving intervals. In particular, we are only interested in the uptrend intervals since the uptrend intervals account for the dispersion process of pollutant particles and enable us to track the process. The air pollutant particles do not take any fixed paths. Therefore, it requires us to select the most likely propagation paths among the monitoring sensors. To achieve this, we propose to compute the causal strengths among the monitoring sensors, which is the task of the "Model Spatio-temporal Causality" component. Both the extracted uptrend intervals and the calculated causal strengths are used to construct the causality graphs to recover the most likely propagation process. Next, we perform the frequent subgraphs mining on these causality graphs to find the frequently occurred propagation patterns. We prove that all the constructed causality graphs are directed acyclic graphs. Based on the special property, we extend the existing frequent subgraph mining algorithm [17] to enumerate the subgraphs more efficiently. After we have the frequent pollutant propagation patterns, the source nodes of every distinct frequent subgraph are either potential pollution sources or locations spatially close to the pollution sources.

In Section 3.1, we first present the definition of uptrend intervals and uptrend events, next we briefly review the fluctuation issue of the geo-sensory data, and finally we present the approach to extracting uptrend intervals from the sensors. In Section 3.2, we first analyze the problem of building sequences among uptrend events, and then we present how to calculate the causality strengths among the extracted uptrend events and provide the algorithm of constructing the pollutant propagation graphs based on the causality strengths. The algorithm of discovering frequent propagation patterns is presented in Section 3.4.

3.1 Uptrend Event Extraction

To facilitate the subsequent discussion we present two basic definitions here:

Definition 3.1. Uptrend Interval. Given a sensor s, an uptrend interval is a consecutive subsequence of measurement $\mathcal{I} = \langle s[t_i], s[t_{i+1}], \ldots, s[t_{i+m-1}]\rangle$ and $\forall j \in \{i, i+1, \ldots, i+m-2\}, s[t_{j+1}] - s[t_j] > 0$, where m denotes the length of the subsequence and $t_i, t_{i+1}, \ldots, t_{i+m-1}$ are the timestamps of every measurement in $\mathcal{I}$.

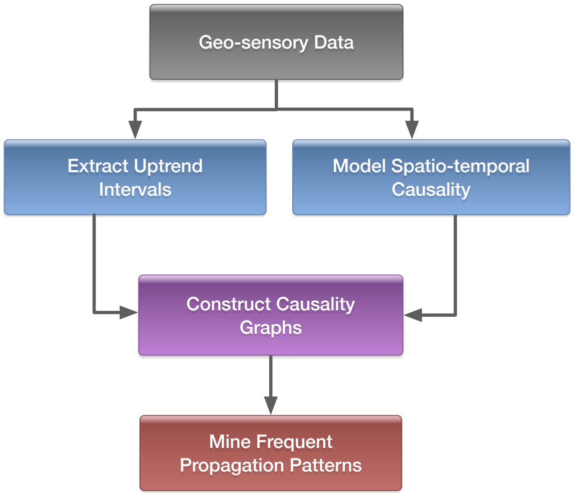

Geo-sensory Data
Extract Uptrend Intervals
Model Spatio-temporal Causality
Construct Causality Graphs
Mine Frequent Propagation Patterns

Figure 2: Framework of our solution

Definition 3.2. Uptrend Event. We associate each uptrend interval $\mathcal{I}$ with an uptrend event e, which is represented by a triple $e = (s, t, v)$, where s is the identifier of the associated sensor, t is the timestamp of the first measurement in $\mathcal{I}$ (i.e., t_i), and v is the value of the first measurement in $\mathcal{I}$ (i.e., $s[t_i]$).

Since the geo-sensory data is typically overwhelmed by various trivial fluctuations, we apply the wavelet transform to capture the multi-resolution evolving intervals by following the previous work [20]. Recall that we aim to discover the pollution sources and pollutant propagation patterns, and thus we are only interested in the emission activities of the sources, which correspond to the uptrend intervals of the geo-sensory data. Consequently, we adopt both wavelet transform and the break and segment strategy [8] to extract the uptrend intervals.

3.2 Spatio-temporal Causality Modeling

After extracting the uptrend events from the geo-sensory data in each individual sensor, we build propagation sequences among them and run frequent pattern discovery algorithm to find frequent propagation patterns. A straightforward method could be to build *causality graphs* [10] where each node indicates an uptrend event and each edge denotes the causality between two events (e.g., $A \rightarrow B$ denotes event A causes event B). Specifically, if A happened before B and the distance between the sensor of A and the sensor of B is smaller than a threshold, we add an edge between A and B. However, such a method is inapplicable for building pollutant propagations due to the following reason. First, there is no explicit propagation path to track, as discussed in Section 1. Second, the propagation of pollutant depends on not only the spatial proximity but also the surrounding POIs (Points of Interest) and various meteorological factors. Third, each sensor may be affected by several pollution sources. As a result, if we use the straightforward method, many redundant edges will be generated. Consequently, we need to address the following problem: How to characterize the causality between two events? Without loss of generality, we assume that event A precedes event B, we represent the sensors of A and B by j and i, respectively, and we use l to denote the time-lag between A and B ($l = B.t - A.t > 0$). To characterize the causality between A and B, we propose to model their causal strength $\beta^{(i)}_{l,j}$ as follows.

Modeling Causal Strength: A widely used model for time series analysis in econometrics is called "Granger Causality" [11], which is introduced by the Nobel prize winning economist, Cilve Granger, and has proven useful in many fields. Granger Causality is based on the premise that "a cause necessarily precedes its effect". Formally, given two time series data $\{x_t\}_{t=1}^T$ and $\{y_t\}_{t=1}^T$, x is said to "Granger cause" y if regressing for y in terms of the past values of y and x is more accurate than regressing just with past values of y,

$$y_t \approx \sum_{l=1}^{L} a_l \cdot y_{t-l} + \sum_{l=1}^{L} b_l \cdot x_{t-l},$$

$$y_t \approx \sum_{l=1}^{L} a_l \cdot y_{t-l},$$

where l is the time lag and L is the maximum time lag we consider.

This motivates us to calculate the causal strengths (β) among events from their historical data. We denote the time series data of sensor as x, its sensor id as i, the number of sensors as N. We calculate β by solving a Elastic Net optimization problem [24] as follows:

$$\beta^{(i)} = \underset{\beta^{(i)}}{\arg\min} \sum_{t=L+1}^{T} (x_{t,i} - \sum_{l=1}^{L} \sum_{j=1, j\neq i}^{N} \beta^{(i)}_{l,j} x_{t-l,j})^2 + \lambda_1 \sum_{l=1}^{L} (\beta^{(i)}_{l,:})^T P \beta^{(i)}_{l,:} + \lambda_2 \|\beta^{(i)}_{:,:}\|_1, \quad (1)$$

where $\beta^{(i)}_{l,j}$ is the causal strength from sensor j to sensor i with a time lag l, since we only intend to learn the causal strength between distinct sensors thus we constrain $j \neq i$. A large $\beta^{(i)}_{l,j}$ indicates a strong causal strength from sensor j to sensor i with a time lag l. The physical meaning of l can be treated as the typical time that pollutant transits from sensor j to sensor i. Here L is the maximum time lag we consider, and P is a diagonal matrix used to enforce the distance-based coefficient decay. In our study the entry corresponding to $\beta_{l,j}$ of P is set by the Euclidean distance between sensor i and j. Finally, $x_{t,j}$ means the measurement data at time t of sensor j.

It is worthwhile mentioning that we do not need to explicitly set a spatial distance to specify which sensors are spatially correlated with sensor i. Instead we use the ℓ_1 regularization term $\|\beta^{(i)}_{:,:}\|_1$ to penalize the corresponding coefficient $\beta_{l,j}$, which would implicitly lead to zero value for the sensors that are spatially uncorrelated with the sensor i. Additionally, the β we calculate is also able to reflect the influence of meteorology. If the wind typically blows from sensor j to i taking l time units, the coefficient $\beta^{(i)}_{l,j}$ tends to have a large magnitude. Note that $\beta^{(i)}$ is a matrix with dimension $L \times (N-1)$, $\beta^{(i)}_{l,:} = \text{vec}(\beta^{(i)}_{l,j})_{j=(1,\ldots,N),j\neq i}$, $\beta^{(i)}_{:,:} = \text{vec}(\beta^{(i)}_{l,j})_{l=(1,\ldots,L),j=(1,\ldots,N),j\neq i}$.

3.3 Causality Graphs Construction

We are now ready to define the concept of event spatio-temporal causality.

Definition 3.3. Event Causality. Given two uptrend events A and B, we say events A, B satisfy event causality or B is caused by A if and only if the following conditions are satisfied: 1) $0 < B.t - A.t \leq L$; 2) denote $j = A.s$, $i = B.s$, $l = B.t - A.t$ the causal strength $\beta^{(i)}_{l,j} \geq \epsilon$ where L is the maximum time lag we consider, ϵ is the minimum threshold.

Note that in Definition 3.3, condition 1) restricts the time continuity between events A and B; condition 2) states that there exists causal strength between event A and B only if the causal strength $\beta^{(i)}_{l,j}$ is a nontrivial value. For convenience, we refer to event A as the causal parent of event B and event B as the causal child of event A.

Definition 3.4. Causality Graph. A causality graph is a directed graph in which each node is associated with a unique uptrend event and the node contains all the fields of the event. An edge $\langle a, b\rangle$ exists in the graph if and only if their associated events A, B satisfy the Definition 3.3.

Definition 3.4 provides us the guide to build causal interactions among the extracted events. The intuition is that given an uptrend event we manage to search all its causal parent events and add an edge from the causal parent event to it, as shown in Algorithm `BuildGraphs`.

Algorithm 1: BuildGraphs

```
   Input: Events $E$ (sorted based on occurring time)
   Output: Causality Graphs $\mathcal{G}$
1  $\mathcal{G} \leftarrow \emptyset$ ;
2  for each event $e \in E$ do
3      $P \leftarrow \emptyset$, $Q \leftarrow \emptyset$;
4      for each graph $g \in \mathcal{G}$ do
5          if InsertNode($g$, $g$.source, $e$) = True then
6              $P \leftarrow P \cup g$;
7          else
8              $Q \leftarrow Q \cup g$;
9      if $P = \emptyset$ then
10         $g' \leftarrow$ Create a new graph with event $e$ as node;
11         $\mathcal{G} \leftarrow Q \cup g'$;
12     else
13         $g' \leftarrow$ Compose the graphs in $P$ based on their common node created
           using e;
14         $g' \leftarrow$ UpdateMaxTimestamp($g'$, $g'$.source);
15         $\mathcal{G} \leftarrow Q \cup g'$;
16 return $\mathcal{G}$;
```

Since the parent event occurred before the child event, we scan the extracted events in chronological order. In lines 4–8 of Algorithm `BuildGraphs` the pre-built graphs g are divided into two groups (P and Q) based on whether the current event can be inserted into the graphs. If no parent event is found in $\mathcal{G}$, we create

Algorithm 2: InsertNode

```
   Input: Causality Graph $g$, Node $n$, Threshold $L$, $\epsilon$, Event $e$
   Output: Boolean Value True or False
1  if $e.t - n.MaxTimestamp > L$ then
2      return False;
3  for $a \in$ Ancesters of node $n$ do
4      if $a.s = n.s$ then
5          return False;
6  success ← False;
7  $l \leftarrow e.t - n.t$, $i \leftarrow e.s$, $j \leftarrow n.s$;
8  if $0 < l \leq L$ and $\beta^{(i)}_{l,j} \geq \epsilon$ then
9      $n' \leftarrow$ Create a new node with event $e$;
10     $n'.MaxTimestamp \leftarrow n'.t$;
11     Add an edge from node $n$ to $n'$;
12     success ← True;
13 for $s \in$ Successors of node $n$ do
14     if $s$ is unexplored and InsertNode(g, s, e) = True then
15         success ← True;
16 return success;
```

Algorithm 3: UpdateMaxTimestamp

```
   Input: Causality Graph $g$, Node $n$
1  for $s \in$ Successors of node $n$ do
2      UpdateMaxTimestamp($g$, $s$);
3      if $s.MaxTimestamp > n.MaxTimestamp$ then
4          $n.MaxTimestamp \leftarrow s.MaxTimestamp$;
```

a new graph using the event (line 10); otherwise we compose the graphs in which the current event is successfully inserted into one based on their common newly inserted node in order to ensure the event only appears once in the causality graphs (line 13). Figure 3 demonstrates the composition of two graphs based on their common node 5, and a virtual node 0 is added as the source node of the composed graph.

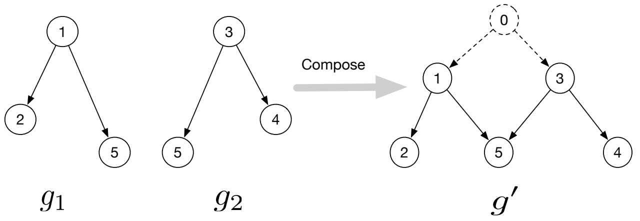


Figure 3: Composing two graphs g_1, g_2 into g'

The InsertNode function checks whether the current node n can be the parent of event e in line 8 and recursively visits the graph g. We do not allow the causality to appear between the events of the same sensor. Therefore we check every ancestor of current node n to eliminate the occurrence of self-dependent in the graph in lines 3–5, i.e., if event A is the parent event of B then $A.s \neq B.s$.

To accelerate the inserting process, we add an additional field named *MaxTimestamp* in the current node, which represents the maximum timestamp in the nodes which are reachable from the current node. We initialize *MaxTimestamp* with the event's occurring time when the node is created, and we update it once a new node is inserted successfully in Algorithm UpdateMaxTimestamp. In line 1 of InsertNode once we find $e.t - n.MaxTimestamp > L$, we can safely stop the insertion since the occurring time of all the descendants of node n is less than $n.MaxTimestamp$.

Theorem 3.5. *The causality graphs built by algorithm* BuildGraphs *are Directed Acyclic Graphs.*

Proof. Based on definitions 3.3 and 3.4, for any path $\langle n_i, n_{i+1}, \ldots, n_{i+m}\rangle$, $i \geq 0$, $m \geq 1$ in the causality graph we have $n_i.v > n_{i+1}.v > \ldots, n_{i+m}.v$. Therefore no cycle exists in the path. □

Theorem 3.5 states that all the constructed causality graphs are DAG (Directed Acyclic Graph). The property is crucial for us to develop efficient frequent propagation patterns mining algorithm. This enables us to extend the gSpan algorithm [17] to more efficiently enumerate the substructures. We will elaborate the details in Section 3.4.

3.4 Frequent Propagation Patterns Mining and Pollution Source Detection

3.4.1 Frequent Propagation Patterns Mining. Frequent subgraphs mining has been extensively studied during the past decades [17, 18, 25], and gSpan [17] is one of the most well-known algorithms. Algorithm gSpan uses the rightmost extension strategy to efficiently enumerate the subgraphs. However, gSpan is a general mining algorithm designed for undirected graphs in which the edges are only distinguished between tree edges and nontree edges. In this subsection, we design a strategy based on the rightmost extension, which fits better with directed acyclic graphs.

Definition 3.6. Tree Edge, Forward Edge and Cross Edge. During a Depth-First Search (DFS), if v is visited for the first time as we traverse the edge $\langle u, v\rangle$, then the edge is a **Tree Edge**. **Forward Edge** is an edge that leads from a node to a nonchild descendant node in the DFS tree and **Cross Edge** is an edge that leads to neither a descendant nor an ancestor node [4, 5].

Figure 4 shows the types of edge in a directed graph, where $\langle a, b\rangle$, $\langle b, c\rangle$ and $\langle b, d\rangle$ are tree edges in a DFS traverse while $\langle a, d\rangle$ is forward edge and $\langle d, c\rangle$ is a cross edge.

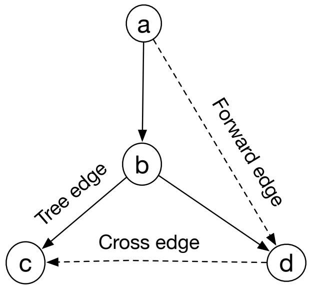


Figure 4: Tree Edge, Forward Edge and Cross Edge

Since the causality graph is a directed acyclic graph, there are no backward edges and it contains only tree edges, forward edges and cross edges. Recall that each node in the causality graph has a field s (sensor id). If we traverse the causality graph in DFS and visit the successors of current node based on the order of their sensor id, we can guarantee the generated DFS code is the minimum DFS code [17]. Thus, an edge $\langle u, v\rangle$ is allowed to grow from the current vertex u if and only if:

(1) $\langle u, v\rangle$ is a tree edge and u is the rightmost vertex, and no successor of u in current DFS code is larger than v, or
(2) $\langle u, v\rangle$ is a forward or cross edge and u is on the rightmost path, and no successor of u in current DFS code is larger than v

```
Algorithm 4: FrequentPatterns
  Input: Causality Graphs $\mathcal{G}$, Minimum Support $minSup$
  Output: Frequent Patterns
  $frequentEdges \leftarrow$ filter out all frequent edges in $\mathcal{G}$;
  $frequentPatterns \leftarrow \emptyset$;
  for each edge $e \in frequentEdges$ do
      $frequentPatterns \leftarrow frequentPatterns \cup$ SubgraphMining($\mathcal{G}$, $e$, $minSup$);
  return $frequentPatterns$
```

```
Algorithm 5: SubgraphMining
  Input: Causality Graphs $\mathcal{G}$, DFS Code $s$, Minimum Support $minSup$
  Output: Frequent Patterns
  $frequentPatterns \leftarrow frequentPatterns \cup s$;
  Enumerate $s$ in each graph in $\mathcal{G}$ and perform rightmost extension based on the aforementioned strategy;
  $C \leftarrow$ all frequent children of $s$;
  for each child $c \in C$ do
      SubgraphMing($\mathcal{G}$, $c$, $minSup$)
  return $frequentPatterns$
```

The overall procedure of Algorithm FrequentPatterns is similar to gSpan except that we extend the rightmost extension strategy to fit better with the DAG. Note that all the DFS codes generated by our strategy is guaranteed to be the minimum DFS code, and thus there is no need to prune the non-minimum DFS code in Algorithm SubgraphMining.

3.4.2 Pollution Source Detection. Each distinct frequent subgraph returned by our algorithm represents a propagation pattern. As we enumerate the subgraphs based on the rightmost extension strategy, each discovered frequent subgraph will only have one source node. For these source nodes in the mined frequent subgraphs, we either treat them as pollution sources or as pollution source proxies (i.e., the locations of the sensors of these source nodes are closer to the pollution sources in comparison to the other sensors in the discovered subgraphs).

4 EXPERIMENTS AND ANALYSIS

In this section, we evaluate the efficacy and efficiency of our solution as well as analyze the experimental results. Firstly, the dataset details are described in subsection 4.1 and then we design a temporal prediction experiment to verify the discovered pollution sources and propagation patterns in subsection 4.2. The efficiency and the analysis of building the causality graphs are discussed in subsection 4.3. Finally, we present the case study in 4.4. All our experiments are conducted in a PC with 3.2 GHZ CPU and 16GB memory.

Table 1: Evaluation on the temporal prediction result.

Prediction Methods	Normal Instances		Sudden Changes	
	p	e	p	e
LR	0.684	27.5	0.298	103.2
ANN	0.646	29.9	0.221	73.7
ANN+KNN	0.667	28.1	0.305	70.2
ANN+Propagation	**0.713**	**21.2**	**0.322**	**50.4**

4.1 Data Description

We utilize real air quality monitoring datasets collected in Beijing. The dataset contains the real-valued AQI (Air Quality Index) of two kinds of pollutants, $PM_{2.5}$ and PM_{10}. The first dataset, *AQRH*, is collected by 204 monitors densely deployed in Haidian district of Beijing from May 27, 2015 to October 1, 2015 and the second dataset, *AQRB*, is collected by 157 monitors deployed in the whole city of Beijing from October 5, 2015 to February 1, 2017. As shown in Figure 5, the left illustrates the monitors' deployment of AQRH and the right illustrates the monitors' deployment of AQRB. All the monitors report their measurement hourly.

We extract 11,532 uptrend events from the dataset *AQRH* and 34,223 uptrend events from the dataset *AQRB* using the method presented in Section 3.1 with the minimum length $m = 5$ (hours).

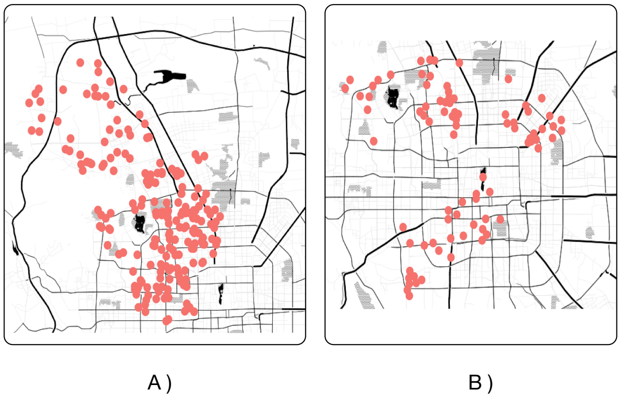


Figure 5: Deployment map of two datasets. The left shows the positions of the 204 sensors deployed in Haidian district and the right shows the positions of the 157 sensors in the urban area of Beijing.

4.2 Evaluation on the Temporal Prediction Task

4.2.1 Experimental Design. It is very challenging to directly verify the effctiveness of the proposed method as we lack the ground truth data (i.e., the true pollution sources). To circumvent it, we design a temporal prediction experiment in which we attempt to predict sensor values for the next six hours with the historical values of the sensors. The rational is that the true pollution sources and propagation patterns should be helpful in making the temporal prediction. To make the prediction, we employ the following methods: LR (Linear Regression), ANN (Artificial Neural Network), ANN+KNN (ANN + K-Nearest Neighbours) and ANN+Propagation (ANN+ mined Propagation patterns). In LR and ANN we make prediction only with the historical data of the sensor itself; In ANN+KNN we use the historical data of both the sensor itself and its k-nearest neighbours; for ANN+Propagation we use the historical data of both the sensor itself and all the ancestors of the sensor in the mined propagation pattern. We measure the mean accuracy (p) and mean absolute error (e) between the predicted value $\widehat{y_i}$ and the ground truth value y_i according to Equation 2 and 3 respectively, where n is the number of instances measured in the next six hours.

$$p = 1 - \frac{\sum_i^n |\widehat{y_i} - y_i|}{\sum_i^n y_i} \tag{2}$$

$$e = \frac{\sum_i^n |\widehat{y_i} - y_i|}{n} \tag{3}$$

4.2.2 The Evaluation Results. Table 1 shows the prediction results of the different methods, in which Normal Instances (Sudden Changes) represent the future sensor values in the next six hours that change less (more) than 150 against its current sensor values. For all instances, we use their past one week historical values in the prediction. In general, LR has a similar performance with ANN in predicting normal instances but less effective than ANN in dealing with the sudden changes. ANN + KNN makes a performance increase in the prediction when feeding with the readings of both the local and surrounding sensors. The results presented in Table 1 show the advantages of the ANN + Propagation which uses the readings of both the local sensors and all ancestor sensors of the local sensors in the mined propagation patterns to make prediction. It achieves a big improvement in the overall accuracy, especially in the scenarios of sudden change, which demonstrates that the propagation patterns found by our method indeed benefit the temporal prediction.

4.3 Evaluation of Building Causality Graphs

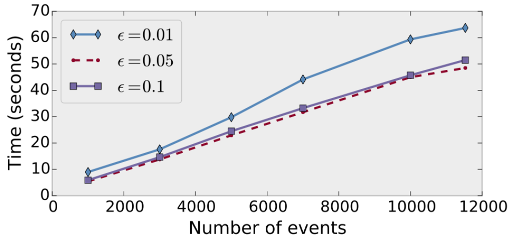


Figure 6: Average running time of building causality graphs with different ϵ.

4.3.1 Efficiency of Building Causality Graphs. Figure 6 shows the time cost of building causality graphs. We can see that the time cost increases linearly as we increase the number of uptrend events. The reason is that the time complexity of building causality graphs depends on the maximum depth in each causality graph and the number of causality graphs. Note that the maximum depth and the number of causality graph rely on the threshold ϵ and the number of events. Moreover, based on Definition 3.3, when we decrease ϵ or increase the number of uptrend events, the number of valid edges will increase, which incurs a higher time complexity. Additionally, it is worthwhile mentioning that the time cost decreases when we increase ϵ from 0.01 to 0.05, but it slightly increases as we further increase ϵ from 0.05 to 0.1. The reason is that although the maximum depth of the causality graphs we built decreases, the number of causality graphs increases, as suggested in Table 2.

Table 2: Statistics of the Built Causality Graphs

ϵ	#Graphs	Total #edges	Average #edges	Maximum #edges
0.01	3705	19523	5.3	959
0.05	6018	14742	2.5	341
0.1	7723	12879	1.7	214

4.3.2 Result of the Built Causality Graphs. Table 2 demonstrates the statistics of the constructed causality graphs on dataset AQRH, which includes the number of causality graphs, the total and average number of edges, and the maximum number of edges in one causality graph. As we increase ϵ, the number of graphs increases and the number of edges decreases. In addition, the average number of edges is relatively small because a large fraction of graphs is just a "single-node graph", and thus we filter out the small graphs (with the number of nodes smaller than 5) as they cannot represent meaningful patterns in our subsequent process (i.e., mining propagation patterns).

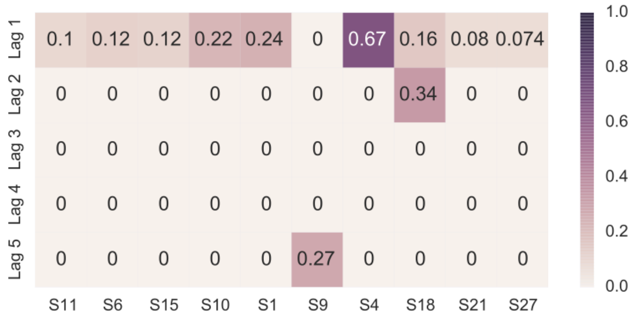
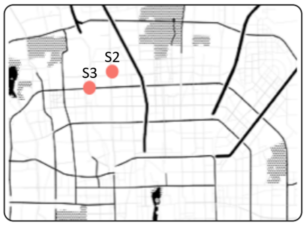


Figure 7: The heatmap of β of sensor s_5 in Haidian, we select the top 10 sensors which have the most significant influence on the sensor.

4.3.3 Demonstration of the Computed Causal Strengths. Figure 7 demonstrates the computed values of $\beta^{(5)}$. We select top-10 sensors that have the largest causal strength values in $\beta^{(5)}$. We can see that most of sensors get their largest causal strength values with 1-hour lag. However, s_9 and s_{18} achieve their largest causal strength values with 5-hour lag and 2-hour lag, respectively.

Two sensors with mutual causal strengths, $\beta^{(2)}_{:,3}$ and $\beta^{(3)}_{:,2}$ are shown in Figure 8. Here $\beta^{(2)}_{1,3}$ contains a relatively significant value



$$\begin{pmatrix} & \beta^{(2)}_{:,3} & \beta^{(3)}_{:,2} \\ \text{Lag}_1: & 0.298 & 2\times10^{-6} \\ \text{Lag}_2: & 6.40\times10^{-7} & 4\times10^{-6} \\ \text{Lag}_3: & 1.30\times10^{-6} & 4\times10^{-6} \\ \text{Lag}_4: & 1.55\times10^{-6} & 3\times10^{-6} \\ \text{Lag}_5: & 1.69\times10^{-6} & 1\times10^{-6} \end{pmatrix}$$

Figure 8: The causal strengths of s_2 and s_3. Here $\beta^{(2)}_{1,3}$ has a significant magnitude while $\beta^{(3)}_{:,2}$ are all trivial values, this indicates s_2 is more likely to being polluted by s_3 but not vice versa.

while all values in $\beta^{(3)}_{:,2}$ are close to 0, which indicates that pollutant is more likely to transit from s_3 to s_2 but not vice versa. This observation demonstrates that it is inappropriate to add an edge between any two uptrend events based on Euclidian distance when constructing the causality graphs.

4.4 Case Study

In this subsection, we first describe the real case studies of the pollution source discovery. Next, we present the propagation path result, system screenshot and user feedback.

After removing the small frequent graphs ($\epsilon = 0.01$) with nodes less than 5, we obtain 216 and 626 graphs for AQRH and AQRB, respectively. In `FrequentPatterns` mining, we set *minSup* at 10.

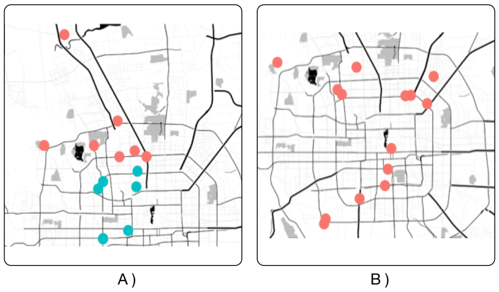


Figure 9: Pollution sources found. A) shows the pollution sources found using the Haidian dataset; B) shows the pollution sources found using the Beijing dataset. Most of the pollution sources are close to the primary roads. The blue points and red points in the left figure indicate the pollution sources discovered within and outside the three-ring Road of Beijing respectively.

4.4.1 Pollution Sources. Figure 9 shows **Pollution Sources** discovered by our proposed paradigm. In both datasets (AQRH and AQRB), most of the pollution sources are located in the key traffic hubs. Such a finding is in accord with the intuition that there is no big factory in the urban area of Beijing and the vehicles may be a huge contributor to the pollution.

There are no overlapping sources between Figure 9-A) and Figure 9-B) because the original monitoring data belong to different regions, as shown in Figure 5. The sources found with the AQRH dataset are more precise than the result discovered with the AQRB dataset since the AQRH dataset is more dense.

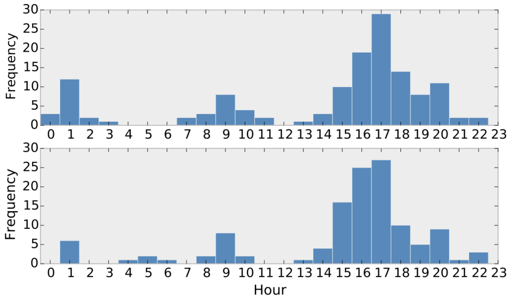


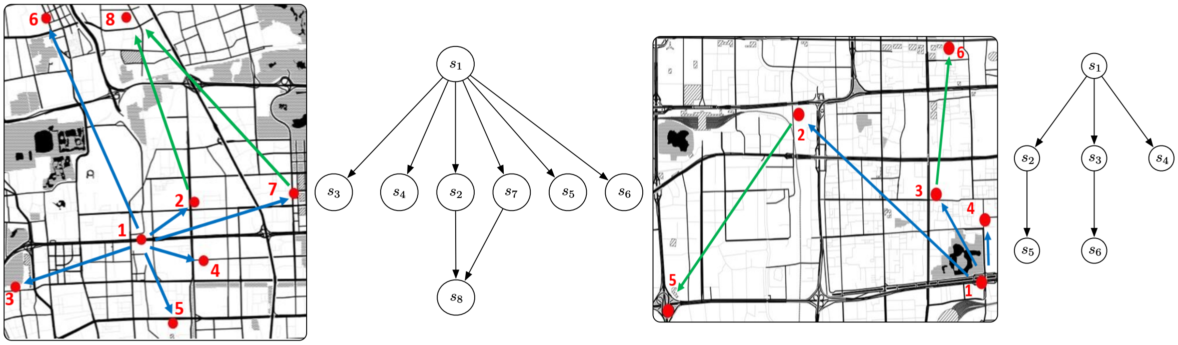
Figure 10: Frequency of starting time of frequent propagation patterns

The ability to find the true pollution sources make our approach suitable to deal with the densely deployed sensors to get the deep understanding of the evolving propagation patterns. In addition, the discovered sources and the propagation patterns provide valuable information for other related domains. For example, we can use the discovered knowledge to improve the accuracy of temporal prediction. It is also helpful in choosing locations for the new air quality monitoring stations and provide suggestions for the environmental control system.

4.4.2 Propagation Path. In this subsection, we report a case study for the pollution propagation.

Figure 10 shows the distribution of the starting time of frequent propagation patterns. It reveals the distinct propagation patterns for three different time intervals, [0, 2], [8, 10], and [15, 20]. The upper part of Figure 10 is the frequency of the starting time of the sources in the suburb area (red sources in Figure 9-a), while the bottom part of Figure 10 corresponds to the sources in the inside urban area (blue sources in Figure 9-a). In the Haidian district, the main pollution sources are emission of vehicles, catering, and coal-fired heating. The time intervals of [8, 10] and [18, 20] belong to the morning and evening rushing hours of traffics, while time interval [15, 20] is also the peak hour of catering. It is worth mentioning that for interval [0, 2], the frequency of the upper part of Figure 10 is significantly higher than that of the bottom one. This discovery coincides with the policy that the heavy vehicles are permitted to enter the suburb areas of Beijing during the night, which produce pollutions.

Figure 11-a and Figure 11-b show examples of two propagation patterns discovered from Haidian district, Beijing, and Fengtai district, Beijing, respectively. Our proposed algorithm can discover multi-layer propagation path and we can show the pollution flow directions directly on the map. In Figure 11-a, s_1 is located in a key traffic hub with catering and the other sensors are located in the schools, which are relatively clean and produce less $PM_{2.5}$. We also compared the result with the Meteorological Data and detect no significant relationship, which indicates that the mined propagation paths might not be induced by the meteorological factors.

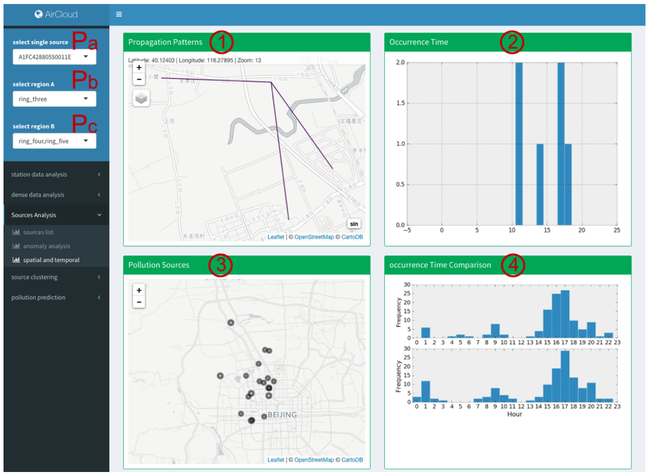


(b) Propagation pattern (Fengtai)

(a) Propagation pattern (Haidian)

Figure 11: Pollution propagation patterns

4.4.3 *Online System and User FeedBack.* The pollution sources and propagation patterns discovered by our method can help the government to make policy in environment protection. Figure 12 shows our online system, where ① and ② is the pollution propagation patterns and pollution occurring time of the selected source P_a respectively; ③ is the sources list in the monitored region; ④ is the histogram of the pollution occurring time of the selected source list P_b and P_c.


select single source: A1FC42880550011E

select region A: ring_three

select region B: ring_four,ring_five

Latitude: 40.12403 | Longitude: 116.27895 | Zoom: 13

Figure 12: The real case systems

The system has been deployed in serveral local EPAs (Environmental Protection Agency) in Beijing[1] and Hebei[2], China, and is used in their daily work. Two of the EPA staffs give their comments to our system:

User A: The system truly assists our daily management and it not only enables the environment protection to become a clear and feasible task but also significantly improves our working efficiency.

User B: The system locates the pollution sources accurately, which really benefits the environmental governance and has made a remarkable achievement in improving the living environment.

5 CONCLUSIONS

In this paper, we study the problem of discovering the potential pollution sources and pollution propagation patterns based on the densely deployed monitoring sensors. We propose to use causal strengths to model the spatio-temporal uptrend events, based on which we construct the causality graphs to illustrate the potential propagations. Finally, we tailor an existing frequent subgraph mining algorithm to mine the propagation patterns more efficiently.

We evaluate our proposed approach using two real world data collected from densely deployment sensors and verify the found potential pollution sources and propagation patterns on the temporal prediction task to show the effectiveness of the paradigm. Compared with the naïve pollution discovery method, our method have the potential to find the true pollution sources and the underlying propagation patterns, which is valuable to have a deep understanding of the air quality evolvement and provides a novel tool to improve the accuracy of the temporal prediction problem. The discovered sources and evolving patterns can also help make location recommendation for the new monitoring stations.

For the future work, we plan to leverage the discovered results to make forecast and perform a thorough study on the causal relationship between the discovered results and other factors, such as meteorological factors, nearby POIs, or traffic context.

Acknowledgements. This work is partly supported by the Rapid-Rich Object Search (ROSE) Lab at Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office. We would like to thank Feng Shanshan for his help in making the figures used in the paper. We would also like to thank to the Haidian Environment Protection Agency staff in Beijing, for their participation and thoughtful comments that vastly improved the quality of this work.

[1] http://hb.bjsjs.gov.cn, http://cphbj.bjchp.gov.cn, http://hdhbj.bjhd.gov.cn

[2] http://www.qhdhb.gov.cn

REFERENCES

[1] Yuyu Chen, Avraham Ebenstein, Michael Greenstone, and Hongbin Li. 2013. Evidence on the impact of sustained exposure to air pollution on life expectancy from Chinafis Huai River policy. *Proceedings of the National Academy of Sciences* 110, 32 (2013), 12936–12941.

[2] Yun Cheng, Xiucheng Li, and Yan Li. 2016. Finding Dynamic Co-evolving Zones in Spatial-Temporal Time Series Data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 129–144.

[3] Yun Cheng, Xiucheng Li, Zhijun Li, Shouxu Jiang, Yilong Li, Ji Jia, and Xiaofan Jiang. 2014. Aircloud: A cloud-based air-quality monitoring system for everyone. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. 251–265.

[4] Thomas H Cormen. 2009. *Introduction to algorithms.* MIT press.

[5] Sanjoy Dasgupta, Christos H Papadimitriou, and Umesh Vazirani. 2006. *Algorithms.* McGraw-Hill, Inc.

[6] Clive WJ Granger. 1980. Testing for causality: a personal viewpoint. *Journal of Economic Dynamics and control* 2 (1980), 329–352.

[7] Haiming Jin, Lu Su, Danyang Chen, Klara Nahrstedt, and Jinhui Xu. 2015. Quality of information aware incentive mechanisms for mobile crowd sensing systems. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 167–176.

[8] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. 2004. Segmenting time series: A survey and novel approach. *Data mining in time series databases* 57 (2004), 1–22.

[9] Yuncheng Li, Jifei Huang, and Jiebo Luo. 2015. Using user generated online photos to estimate and monitor air pollution in major cities. In *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*. ACM, 79.

[10] Wei Liu, Yu Zheng, Sanjay Chawla, Jing Yuan, and Xie Xing. 2011. Discovering spatio-temporal causal interactions in traffic data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1010–1018.

[11] Aurelie C Lozano, Hongfei Li, Alexandru Niculescu-Mizil, Yan Liu, Claudia Perlich, Jonathan Hosking, and Naoki Abe. 2009. Spatial-temporal causal modeling for climate change attribution. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 587–596.

[12] XH Ma, Lu Gan, AY Zhang, NJ Li, and MY Zhang. 2013. Cause analysis on durative fog and haze in January 2013 over Beijing area. *Adv Environ Prot* 3 (2013), 29–33.

[13] Chuishi Meng, Wenjun Jiang, Yaliang Li, Jing Gao, Lu Su, Hu Ding, and Yun Cheng. 2015. Truth discovery on crowd sensing of correlated entities. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 169–182.

[14] Chuishi Meng, Houping Xiao, Lu Su, and Yun Cheng. 2016. Tackling the Redundancy and Sparsity in Crowd Sensing Applications. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 150–163.

[15] Jingbo Shang, Yu Zheng, Wenzhu Tong, Eric Chang, and Yong Yu. 2014. Inferring gas consumption and pollution emission of vehicles throughout a city. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1027–1036.

[16] Ke Wang and Yingnan Liu. 2014. Can Beijing fight with haze? Lessons can be learned from London and Los Angeles. *Natural hazards* 72, 2 (2014), 1265.

[17] Xifeng Yan and Jiawei Han. 2002. gspan: Graph-based substructure pattern mining. In *IEEE International Conference on Data Mining (ICDM)*. 721–724.

[18] Xifeng Yan and Jiawei Han. 2003. CloseGraph: mining closed frequent graph patterns. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 286–295.

[19] Mohammed J Zaki. 2002. Efficiently mining frequent trees in a forest. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 71–80.

[20] Chao Zhang, Yu Zheng, Xiuli Ma, and Jiawei Han. 2015. Assembler: Efficient Discovery of Spatial Co-evolving Patterns in Massive Geo-sensory Data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1415–1424.

[21] Dongyong Zhang, Junjuan Liu, and Bingjun Li. 2014. Tackling air pollution in Chinafi!?What do we learn from the great smog of 1950s in London. *Sustainability* 6, 8 (2014), 5322–5338.

[22] Quan Zhou, Wenlin Chen, Shiji Song, Jacob R Gardner, Kilian Q Weinberger, and Yixin Chen. 2014. A reduction of the elastic net to support vector machines with an application to gpu computing. *arXiv preprint arXiv:1409.1976* (2014).

[23] Julie Yixuan Zhu, Yu Zheng, Xiuwen Yi, and Victor OK Li. 2016. A Gaussian Bayesian model to identify spatio-temporal causalities for air pollution based on urban big data. In *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*. IEEE, 3–8.

[24] Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67, 2 (2005), 301–320.

[25] Zhaonian Zou, Jianzhong Li, Hong Gao, and Shuo Zhang. 2009. Frequent subgraph pattern mining on uncertain graph data. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 583–592.