Multidimensional Analysis of Atypical Events in Cyber-Physical Data

Lu-An Tang[1], Xiao Yu[1], Sangkyum Kim[1], Jiawei Han[1]
Wen-Chih Peng[2], Yizhou Sun[1], Hector Gonzalez[3], Sebastian Seith[4]
[1]University of Illinois at Urbana-Champaign
[2]National Chiao Tung University, [3]Google Research, [4]Morningstar Inc.
{tang18, xiaoyu1, kim71, hanj, sun22}@illinois.edu, wcpeng@cs.nctu.edu.tw
hagonzal@google.com, sebastian.seith@morningstar.com

***Abstract*— A Cyber-Physical System (CPS) integrates physical devices (*e.g.*, sensors, cameras) with cyber components to form a situation-integrated analytical system that may respond intelligently to dynamic changes of the real-world situations. CPS claims many promising applications, such as traffic observation, battlefield surveillance and sensor-network-based monitoring. One important research topic in CPS is about the atypical event analysis, *i.e.*, retrieving the events from a large amount of data and analyzing them with spatial, temporal and other multi-dimensional information. Many traditional approaches are not feasible for such analysis since they cannot describe the complex atypical events. In this study, we propose a new model of the *atypical cluster* to effectively represent such events and efficiently retrieve them from massive data. The micro-cluster is designed to summarize an individual event, and the macro-cluster is used to integrate the information from multiple events. To facilitate scalable, flexible and online analysis, the concept of *significant cluster* is defined and a guided clustering algorithm is proposed to retrieve significant clusters in an efficient manner. We conduct experiments on real datasets with the size of more than 50 GB. The results show that the proposed method can provide more accurate information with only 15% to 20% time cost of the baselines.**

I. Introduction

The *Cyber-Physical Systems* (CPS) has been a focused research theme in recent years. It is placed on the top of the priority list for federal research investment in the fiscal year report of U.S. President's council of advisors on science and technology [3]. A CPS consists of a large number of sensors and collects huge amount of data with the information of locations, time, weather, temperature and so on. They have wide applications in the areas of traffic monitoring, battlefield surveillance and sensor-network-based monitoring [5], [11], [18].

In CPS applications, some sensors occasionally report unusual or abnormal readings (*i.e.*, atypical data), such data may imply fundamental changes of the monitored objects and possess high domain significance. To benefit the system's performance and help user's decision making, it is important to analyze the atypical data with spatial, temporal and other multi-dimensional information in an integrated manner. A motivation example is shown as follows.

Example 1. The highway traffic monitoring system is a typical CPS application. With the sensor devices installed on road networks, the monitoring system watches the traffic flow of major U.S highways in 24 hours × 7 days and acquires huge volumes of data [14]. In this scenario, one important type of atypical events is the traffic congestion. Some frequent questions asked by the officers of transportation department are: (1) Where do the traffic congestions usually happen in the city? (2) When and how do they start? (3) On which road segment (or time period) is the congestion most serious?

In such queries, the users are not satisfied merely on a database query returned with thousands of records. They demand summarized and analytical information, integrated in the unit of atypical event. The granularity of the results should also be flexible according to the user's requirements: some officers may be only concerned with the information in recent days, whereas others are more interested in the monthly or even yearly report. However, it is hard to support such multi-dimensional analysis of atypical events in CPS data, partly due to following difficulties:

- **Massive Data**: A typical CPS includes hundreds of sensors and each sensor generates data records in every few minutes. The CPS database usually contains giga-bytes, even tera-bytes of data records. The management system is required to process the huge data with high efficiency.
- **Complex Event**: The atypical event is a dynamic process influencing multiple spatial regions. Those spatial regions expand or shrink as time passes by, they may even combine with others or split into smaller ones. Hence the atypical events do not have fixed spatial boundaries. They are difficult to be represented by traditional models.
- **Information Integration**: In many applications, the users demand integrated information for analytical purposes. For example, a transportation officer may need a monthly summary of the congestions in the city. Then the system has to measure the similarity among daily atypical events and integrate the similar ones to provide a general picture.
- **Retrieving Effectiveness**: A large-scale analytical query may contain the data from hundreds of atypical events, however, not all of them are interesting to the users. The users may only prefer a few significant results, *i.e.*, the

most serious events that influence large area and last for a long time. The system should distinguish such significant events in the retrieving process and emphasize them from the majority of trivial ones.

In this study, we introduce the techniques to discover atypical events and summarize them as *atypical clusters*. The atypical cluster is a model describing multi-dimensional features of the atypical event. They can be efficiently integrated in a hierarchical framework to form macro-clusters for large-scale analytical queries. To retrieve significant macro-clusters, the system employs a guided clustering algorithm to filter out the trivial results and meanwhile guarantees the accuracy of significant clusters. To the best of our knowledge, this study is the first one to generate, integrate and retrieve clusters from atypical CPS data for multidimensional analysis. The proposed methods are evaluated on gigabyte-scale datasets from real applications, our approaches can provide more detailed and accurate results with only 15% to 20% time cost of the baselines.

The rest of the paper is organized as follows. Section II introduces the problem formulation and system framework; Section III proposes the models of atypical clusters and the algorithms of cluster integration; Section IV introduces the techniques to efficiently retrieve significant clusters for online queries; Section V evaluates the performances of proposed methods on real datasets; Section VI briefly discusses the related work and in Section VII we make the conclusion.

II. Overview

A. Problem Formulation

The cyber-physical systems monitor real world by sensor networks. In most cases, a sensor reports records with normal readings. If an atypical event happens (such as a congestion is detected in traffic system), the sensor will send out atypical records. The detailed atypical criteria are different according to the application scenarios and environments (*e.g.*, the highway types and speed limits), many state-of-the-art methods have been proposed to select the trustworthy atypical records in traffic, battlefield and other CPS data [19], [17], [22], [18]. Since the main theme of this study is on multi-dimensional analysis of atypical event, we assume that the atypical criteria is given and clean and trustworthy atypical records can be retrieved by CPS. In fact, some of such datasets are available to public [2].

The atypical records are represented in the format of (s, t, $f(s,t)$), where the severity measure $f(s,t)$ is a numerical value collected from sensor s in time window t. Without loss of generality, we adopt the atypical duration as the severity measure in this study, since it is commonly used in many CPS applications. For example, (s_1, 8:05am-8:10am, 4 mins) means that sensor s_1 has reported atypical readings for 4 minutes from 8:05am to 8:10am. Note that, although we focus on atypical duration in this paper, the proposed approach is also flexible to adjust to other domain specific measures.

Some existing methods aggregate the severity measures in a bottom-up style and use the aggregated value to answer the analytical query [15]. They pre-define aggregation hierarchies on temporal, spatial and other related dimensions and accumulate the value of severity measure following such hierarchies. For example, the traffic monitoring system may sum up the congestion duration by hour, day, month and year in temporal dimension. The spatial regions are partitioned by zipcode areas, streets [15], highway mileages [7], or the R-trees rectangles [12]. Since the sensors are usually fixed in their locations, with the help of a topology graph mapping the sensors to different regions, the spatial coverage can be represented by a set of sensors. For an analytical query in region W and time T, the aggregated value of severity measure $F(W,T)$ is then computed as follows.

$$F(W,T) = \sum_{s \in W} \sum_{t \in T} f(s,t) \qquad (1)$$

However, the bottom-up styled aggregation may not satisfy the user's requirements in several cases. This problem can be best understood by an illustrative example as follows.

Example 2. The bottom-up styled aggregation is carried out on zipcode areas. The regions with high severity are tagged out as red zones, *e.g.*, a, b, …, g in Figure 1[1]. However this model only points out where the atypical events are. It does not give detailed information on when those events start and which part is most serious in a specified red zone.

The bottom-up styled method cannot provide details since the numeric measure of severity is too abstract to describe the complex atypical events. In addition, the atypical events may not follow the pre-defined boundaries. There are three major congestion events A, B and C in Figure 1. They are partitioned into seven red zones. The users may think that the fragments of A and B congest together, since they are spatially close to each other. However, a careful examination reveals that the fragment A (freeway 10W) usually congests in the morning rush hours and the fragment B (freeway 10E) jams in the evenings. They seldom congest together in the same time. In such case, only few experienced users that carry out the aggregation by road segments and specify certain temporal constraints (*e.g.*, aggregate the severities by time zone) can distinguish those congestion events. But most users cannot provide such detailed query information in advance. In contrast, the users prefer the system to automatically retrieve atypical events and organize them in a natural and precise format.

Task Specification. Let R be the CPS dataset, the tasks of atypical event analysis are: (1) Retrieving the atypical events from R, representing them with a succinct model and constructing a data structure to store and integrate the information of multiple events; (2) For an analytical query $Q(W,T)$, where W is a spatial region and T is a time period, processing query Q in real time, the query results should include the accurate spatial and temporal information of major atypical events happened in region W during time T.

[1] Figures 1, 7, 11 and 12 in this paper are generated based on Google Map API.

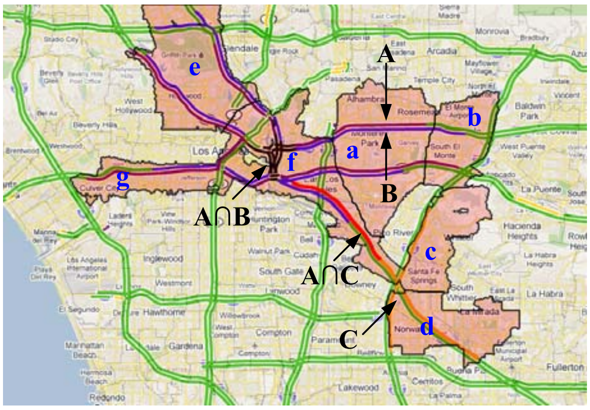


Fig. 1. Example: Problems of Bottom-up Styled Method

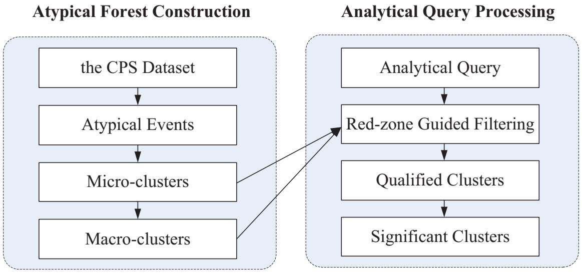


Fig. 2. The Overview of System Framework

B. System Framework

Figure 2 shows the overview of our system framework. The system consists of two components: the atypical forest construction module and the analytical query processing module.

Atypical Forest Construction: This component offline builds up the model of atypical forest from the CPS dataset. The system first retrieves the atypical events from the dataset, then constructs the atypical micro-cluster to store the features of each individual event. The similarity of micro-clusters is measured based on the retrieved features. The system merges similar micro-clusters as macro-clusters to integrate multiple events. The clusters are formed in hierarchical trees to construct the atypical forest, which will be used to help process the analytical queries.

Analytical Query Processing: This component online answers a user specified query. The key issue is to efficiently retrieve significant clusters in the query range. The query processing algorithm first determines the possible regions where the significant clusters might be (*i.e.*, red-zones), then prunes the micro-clusters locating outside those regions. Only the qualified micro-clusters are selected to generate the macro-clusters as query results.

We will introduce the model construction methods in Section III and query processing techniques in Section IV. Figure 3 lists the notations used throughout this paper.

Notation	Explanation	Notation	Explanation
R	the CPS dataset	r_i, r_j	the atypical data records
S	the sensor set	s_1, s_2	the sensors
T	the time period	t_1, t_2	the time windows
SF	the spatial feature	TF	the temporal feature
$f(s, t)$	the severity measure	$F(S, T)$	the total severity
E_A, E_B	the atypical events	W	the spatial region
$Q(W, T)$	the analytical query	C_A, C_B	the atypical clusters
μ_i	the agg. severity by s_i	ν_j	the agg. severity by t_j
δ_t	the time threshold	δ_d	the distance threshold
δ_s	the severity threshold	δ_{sim}	the similarity threshold

Fig. 3. The List of Notations

III. Atypical Forest Construction

A. Atypical Event

The atypical event is a dynamic process including many atypical records. In the traffic system, the atypical event of a congestion usually starts from a single street, which can only be detected by one or few sensors. Then the congestion swiftly expands along the street and influences nearby sensors. A serious congestion usually lasts for a few hours and covers hundreds of sensors when reaching the full size.

By observing the phenomenon of congestion event, we find that those records in an atypical event are spatially close and timely relevant to each other. Hence we introduce the following definitions:

Definition 1 (Direct Atypical Related). Let $r_i\langle s_i,\ t_i,\ f(s_i, t_i)\rangle$ and $r_j\langle s_j,\ t_j,\ f(s_j, t_j)\rangle$ be two atypical records, δ_d be the distance threshold and δ_t be the time interval threshold. r_i and r_j are *direct atypical related* if distance(s_i, s_j) $< \delta_d$ and interval(t_i, t_j) $< \delta_t$.

Definition 2 (Atypical Related). Let r_1 and r_n be two atypical records. If there is a chain of records $r_1, r_2, \ldots, r_n$, such that r_i and r_{i+1} are direct atypical related; then r_1 and r_n are atypical related.

Based on the above concepts, we formally define the atypical event as follows.

Definition 3 (Atypical Event). Let R be the CPS dataset. *Atypical event* E is a subset of R satisfying the following conditions: (1) $\forall r \in E$, r is an atypical record; (2) $\forall$ r_i, r_j: if $r_i \in E$ and r_j is atypical related with r_i, then $r_j \in E$; (3) $\forall$ r_i, $r_j \in E$: r_i is atypical related with r_j.

Example 3. Figure 4 shows three atypical events. Since the events contain hundreds of atypical records, we only list part of their records in the table.

Property 1. The atypical event is a holistic model[2].

B. Atypical Micro-clusters

The atypical event is not feasible to process analytical queries, because the holistic model is inefficient to aggregate

[2] The proofs of all properties in this paper are listed in the Appendix.

ID	Atypical Records
E_A	<s_1, 8:05am - 8:10am, 4 min>; <s_1, 8:10am - 8:15am, 5 min>; <s_2, 8:10am - 8:15am, 5 min>; <s_3, 8:15am - 8:20am, 5 min>; <s_4, 8:15am - 8:20am, 2 min>; . . .
E_B	<s_3, 6:20pm - 6:25pm, 2 min>; <s_4, 6:20pm - 6:25pm, 5 min>; <s_1, 6:25pm - 6:30pm, 5 min>; <s_4, 6:25pm - 6:30pm, 5 min>; <s_5, 6:30pm - 6:35pm, 5 min>; . . .
E_C	<s_1, 8:20am - 8:25am, 1 min>; <s_1, 8:25am - 8:30am, 5 min>; <s_9, 8:25am - 8:30am, 5 min>; <s_1, 8:30am - 8:35am, 5 min>; <s_7, 8:35am - 8:40am, 3 min>; . . .

Fig. 4. Example: The Atypical Events

ID	Spatial Features	Temporal Features
C_A	<s_1, 182 min>; <s_2, 97 min>; <s_3, 33 min>; <s_4, 12 min>; ···	<8:05am - 8:10 am, 4 min>; <8:10 am - 8:15 am, 10 min>;···
C_B	<s_1, 12 min>; <s_2, 51 min>; <s_3, 34 min>; <s_4, 140 min>; ···	<6:20 pm - 6:25 pm, 7 min>; <6:25 pm - 6:30 pm, 13 min>;···
C_C	<s_1, 103 min>; <s_2, 75 min>; <s_7, 54 min>; <s_9, 60 min>; ···	<8:20am - 8:25 am, 1 min>; <8:25 am - 8:30 am, 15 min>;···

Fig. 5. Example: The Atypical Micro-clusters

```
Algorithm 1. Retrieving Micro-clusters
Input: the CPS dataset $R$, the distance threshold $\delta_d$, the tiem
  interval threshold $\delta_t$
Output: the micro-cluster set $micro\_set$.
1.  retrieve the atypical dataset $R_a$ from $R$;
2.  repeat
3.      randomly select an atypical record $r$ from $R_a$;
4.      retrieve atypical related records from $r$ w.r.t $\delta_d$ and $\delta_t$;
5.      group such records as atypical event $E$;
6.      initialize micro-cluster $C<ID, SF, TF>$;
7.      for each sensor $s_i$ of $E$
8.          calculate the sensor severity $\mu_i$;
9.          add $< s_i, \mu_i >$ to $SF$;
10.     for each time window $t_j$ of $E$
11.         calculate the time severity $v_j$;
12.         add $< t_j, v_j >$ to $TF$;
13.     add $C$ to $micro\_set$;
14.     $R_a \leftarrow R_a - E$;
15. until $R_a$ is empty;
16. return $micro\_set$;
```

Fig. 6. Algorithm: Retrieving Micro-clusters from CPS dataset

and compute [9]. Hence a more succinct model is required to describe the atypical events. To this end, we propose the concept of *atypical micro-cluster*.

Definition 4 (Atypical Micro-cluster). Let E be an atypical event with sensor set $S = \{s_1, s_2, \ldots, s_n\}$ and time period $T = \{t_1, t_2, \ldots, t_m\}$, the *atypical micro-cluster* C of E is defined as $C = \langle ID,\ SF,\ TF\rangle$, where ID is the cluster id, the spatial feature SF =$\{\langle s_1,\ \mu_1\rangle,\ \langle s_2,\ \mu_2\rangle,\ \ldots,\ \langle s_n,\ \mu_n\rangle\}$, $\mu_i = \sum_T f(s_i, t)$ is the aggregated severity on sensor s_i; the temporal feature TF =$\{\langle t_1,\ \nu_1\rangle,\ \langle t_2,\ \nu_2\rangle,\ \ldots,\ \langle t_m,\ \nu_m\rangle\}$, $\nu_j = \sum_S f(s, t_j)$ is the aggregated severity during time window t_j.

Intuitively speaking, the spatial feature is the summary of the atypical event by sensors, and the temporal feature is the summary of the event during each time window. μ_i represents how long the sensor s_i is atypical in E and ν_j reflects how many sensors are atypical during time window t_j. In this way, the atypical micro-cluster can represent the coverage, time length and seriousness of corresponding atypical event.

Example 4. Figure 5 shows the atypical micro-clusters retrieved from Example 3. The spatial and temporal features are generated by aggregating the atypical records. Note that, since the sensors may have different atypical durations in a time window, we still use the accumulated time duration to denote the severity in temporal features. The spatial and temporal features can be used to help answer the queries in Example 1, *e.g.*, the atypical event A starts at around 8:05 am and the most serious part is the road segment monitored by s_1, it experiences total 182 minutes of congestion in the event.

The atypical events can be retrieved by a single scan of the dataset and the micro-clusters are generated simultaneously. Algorithm 1 shows the detailed process. The algorithm first scans the CPS dataset as pre-processing step to select the atypical records (Line 1), then randomly picks an atypical record as a seed (Line 3) and retrieves all the related atypical records to form the atypical event (Lines 4 – 5). Then the algorithm summarizes the spatial and temporal features from the atypical event (Lines 7 – 12). The above steps are repeated until all the atypical records are processed (Line 15).

Proposition 1. The time complexity of Algorithm 1 is $O(N+n^2)$ without index and $O(N+n\cdot log(n))$ with index, where N is the size of CPS dataset R and n is the number of atypical records.

Proof: The Algorithm has to scan the dataset and select the atypical records with time N. The major cost of event retrieving is on Line 3 to select the related atypical records. If there is no index on the temporal and spatial dimensions, it costs $O(n)$ time to retrieve related atypical records for a seed. Thus the entire algorithm takes $O(N + n^2)$ time in the worst case. However the searching algorithm can speed up to $O(log(n))$ with indexes and the time complexity of the algorithm is improved as $O(N + n \cdot log(n))$. ■

C. Atypical Cluster Integration

In many scenarios, the users ask integrated information in large query range, *e.g.*, the transportation officers may require a monthly congestion summary of the whole city. And in that month, the 10E freeway often jams near downtown in the evening rush hours. If the system can integrate such similar events into a macro-cluster, it will help user's analysis and decision making.

The first task of cluster integration is to compute the similarities between two atypical clusters. The cluster similarity is measured based on their features. Equations 3 and 4 show the calculation of spatial and temporal similarities, where S_i is the sensor set and T_i is the time window set, μ^i and ν^i are the aggregated severity of sensor s and time window t in cluster C_i, respectively. Equation 3 computes the severity percentages of common sensors over a cluster, and balances the values on two clusters by a mathematical function $g(p_1, p_2)$. The function $g(p_1, p_2)$ could be in the form of max, min, the arithmetic mean, harmonic mean or geometric mean. The reason of using different mathematical balance function here is that the size of two clusters may be different. When comparing the similarity between a large cluster and a small one, the percentage of common sensors is inevitably small for the larger cluster. If we use the max function, the two clusters are still similar even if the common sensor percentage is low for the larger cluster.

$$Sim(C_1, C_2) = \frac{1}{2}(Sim_{SF}(C_1, C_2) + Sim_{TF}(C_1, C_2)) \quad (2)$$

$$Sim_{SF}(C_1, C_2) = g(\frac{\sum\limits_{S_1 \cap S_2} \mu^1}{\sum\limits_{S_1} \mu^1}, \frac{\sum\limits_{S_1 \cap S_2} \mu^2}{\sum\limits_{S_2} \mu^2}) \quad (3)$$

$$Sim_{TF}(C_1, C_2) = g(\frac{\sum\limits_{T_1 \cap T_2} \nu^1}{\sum\limits_{T_1} \nu^1}, \frac{\sum\limits_{T_1 \cap T_2} \nu^2}{\sum\limits_{T_2} \nu^2}) \quad (4)$$

Example 5. Figure 7 shows the three atypical clusters listed in Figure 5. C_A and C_B have several common sensors, including s_1, s_2 and s_3. Their spatial features are similar to each other, however C_A happened in the morning and C_B happened in the evening, they have no common temporal features. Hence they will not be integrated as a macro-cluster. C_A and C_C happened in the similar time and they have many common sensors (*i.e.*, spatially related and timely close), they can be merged to form a macro-cluster.

Once two micro-clusters are merged, a single macro-cluster is created to represent the merge result. The spatial feature of the macro-cluster is calculated as shown in Equation 5: the system accumulates the severities of common sensors from two micro-clusters and keeps the non-overlapping ones, so is the temporal feature (Equation 6). A new ID is generated for the macro-cluster.

$$\begin{aligned} SF_{marco} = \{\langle s_i, \mu_i^1 + \mu_i^2 \rangle | s_i \in (S_1 \cap S_2)\} \\ \cup \{\langle s_j, \mu_j \rangle | s_j \notin (S_1 \cap S_2), s_j \in (S_1 \cup S_2)\} \end{aligned} \quad (5)$$

$$\begin{aligned} TF_{marco} = \{\langle t_i, \nu_i^1 + \nu_i^2 \rangle | t_i \in (T_1 \cap T_2)\} \\ \cup \{\langle t_j, \nu_j \rangle | t_j \notin (T_1 \cap T_2), t_j \in (T_1 \cup T_2)\} \end{aligned} \quad (6)$$

Property 2. The spatial and temporal features in atypical clusters are algebraic.

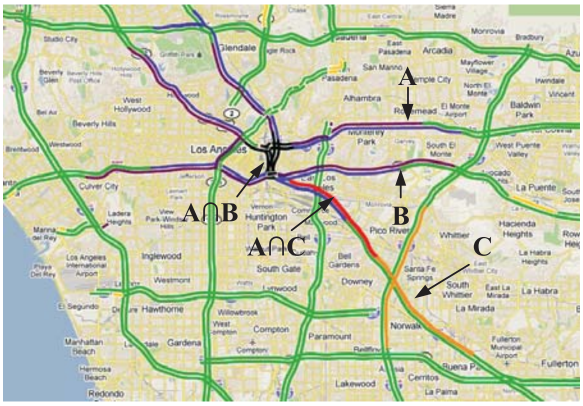
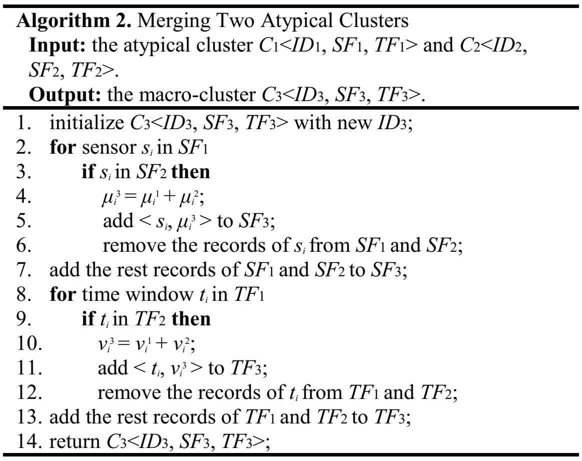


Fig. 7. Example: Retrieving Similar Atypical Clusters



Algorithm 2. Merging Two Atypical Clusters
Input: the atypical cluster C_1<ID_1, SF_1, TF_1> and C_2<ID_2, SF_2, TF_2>.
Output: the macro-cluster C_3<ID_3, SF_3, TF_3>.

```
1.  initialize $C_3$<$ID_3$, $SF_3$, $TF_3$> with new $ID_3$;
2.  for sensor $s_i$ in $SF_1$
3.      if $s_i$ in $SF_2$ then
4.          $\mu_i^3 = \mu_i^1 + \mu_i^2$;
5.          add < $s_i$, $\mu_i^3$ > to $SF_3$;
6.          remove the records of $s_i$ from $SF_1$ and $SF_2$;
7.  add the rest records of $SF_1$ and $SF_2$ to $SF_3$;
8.  for time window $t_i$ in $TF_1$
9.      if $t_i$ in $TF_2$ then
10.         $v_i^3 = v_i^1 + v_i^2$;
11.         add < $t_i$, $v_i^3$ > to $TF_3$;
12.         remove the records of $t_i$ from $TF_1$ and $TF_2$;
13. add the rest records of $TF_1$ and $TF_2$ to $TF_3$;
14. return $C_3$<$ID_3$, $SF_3$, $TF_3$>;
```

Fig. 8. Algorithm: Merging Two Atypical Clusters

The algebraic features are efficient to compute and aggregate [9], thus we use atypical clusters as the major model in the system. The algorithm of merging two atypical clusters are shown in Figure 8. The system initializes the macro-cluster with new ID (Line 1), then accumulates the severity of common sensors in two micro-clusters (Lines 2 – 6) and copies the non-common ones (Line 7); the same steps are carried out in temporal features (Lines 8 – 13).

Proposition 2. Let m_1 and m_2 be the number of sensors in C_1 and C_2, l_1 and l_2 be the number of time windows of C_1 and C_2, the time complexity of Algorithm 2 is $O(m_1 + m_2 + l_1 + l_2)$.

Proof: The system can use a hash map to store the spatial and temporal features, and retrieve a specific feature in constant time. The algorithm has to scan the features once to check the common features. In the worst case, C_1 and C_2 have no common feature, then the system needs to copy all their features, hence the total time complexity of Algorithm 2 is $O(m_1 + m_2 + l_1 + l_2)$. The algorithm's time cost is linear to the total size of the features in C_1 and C_2. ■

```
Algorithm 3. Atypical Cluster Integration
Input: micro-cluster set micro_set, similarity threshold $\delta_{sim}$
Output: the macro-cluster set macro_set
1.  repeat
2.    for each micro-cluster pair $C_1$ and $C_2$
3.        if sim($C_1$, $C_2$)> $\delta_{sim}$ then
4.          $C_3$ = merge ($C_1$, $C_2$); \\Algorithm 2
5.          add $C_3$ to micro_set;
6.          remove $C_1$, $C_2$ from micro_set;
7.  until no clusters can be merged in micro_set;
8.  macro_set ← micro_set;
9.  return macro_set;
```

Fig. 9. Algorithm: Atypical Cluster Integration

Property 3. The operation of merging atypical clusters is mathematically commutative and associative.

Property 3 tells us that the order of micro-clusters does not influence the results of macro-clusters. Thus we design the atypical cluster integration process as shown in Figure 9. The algorithm starts by checking each pair of the micro-clusters. If their similarity is larger than the given threshold, a merge operation is called to integrate them (Lines 2 – 4). This process is irrelevant to the order of micro-clusters. The new cluster is put back to the set and the old pair is discarded (Lines 5 – 6). The program stops until no clusters could be merged (Line 7).

Proposition 3. Let n be the number of micro-clusters, the time complexity of Algorithm 3 is $O(n^2)$.

Proof: The worst case happens when only two micro-clusters can be merged (Phase 1). However, after merging them, the macro-cluster can then be merged with another micro-cluster, and so on, until all the micro-clusters are merged into that macro-cluster (Phase 2). Phase 1 costs $n(n-1)/2$ time to find the merging pair in the worst case, and Phase 2 costs $(n-1+n-2+...+1) = n(n-1)/2$ time. Hence the time complexity of Algorithm 3 is $O(n^2)$. ■

The cluster integration algorithm takes the micro-clusters as input and outputs the macro-clusters to represent the integrated information from multiple atypical events. Such macro-clusters can also be used as new inputs to get even larger clusters. In this way a hierarchical clustering tree is constructed. Figure 10 shows a clustering tree for the traffic congestions. Ten micro-clusters are stored in the lowest level of the tree. The macro-clusters b_1, ..., b_6 are aggregated on them, and they are used to integrate the two larger clusters in the highest level of the tree.

To generate a monthly summary of the congestion events, the system can first retrieve the micro-clusters of each single day and construct the tree in the hierarchy of day-week-month. It may also integrate the micro-clusters by weekdays and weekends. Hence there are multiple clustering trees according to different aggregation paths. Those trees make up a forest of atypical clusters. Such a forest (or parts of it) can be pre-computed to help process the analytical queries.

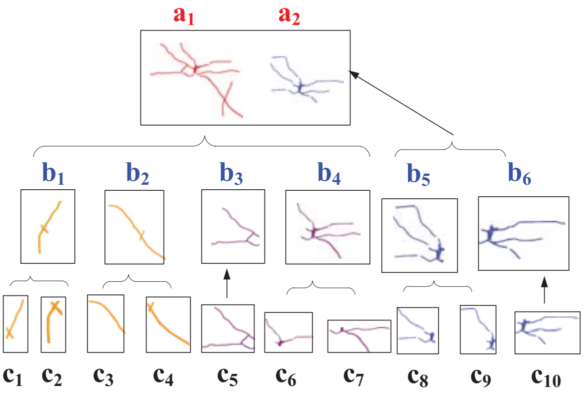


Fig. 10. Example: A Clustering Tree

IV. Analytical Query Processing

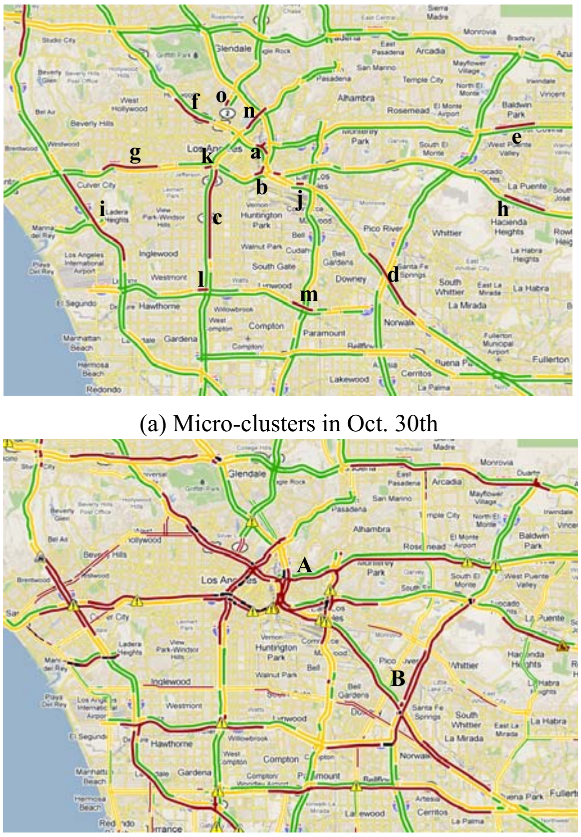
In practical applications we do not pre-compute the entire atypical forest due to storage limits. In most cases only the micro-clusters and some low level macro-clusters are pre-computed. With such a partially materialized data structure, the system needs to dynamically integrate the low-level clusters to process analytical queries with large query range. The online clustering process is similar to the cluster integration algorithm. However there are two problems: (1) Efficiency: the time complexity of cluster aggregation algorithm is quadratic to the number of input clusters. Therefore the system should select only the relevant micro-clusters to reduce the time cost; (2) Effectiveness: If the query scale is large, *e.g.*, the users want the monthly congestion report of the whole city. There may be a large number of macro-clusters in the query range, but only few of them are *significant clusters* with high severities, while the others are negligible. When constructing atypical forest, the process is offline and the system can store all the clustering results. When processing online queries, the users usually demand the significant clusters being delivered in short time and do not prefer the results mixed with trivial clusters.

Definition 5 (Significant Cluster). Let $Q(W, T)$ be a query with the range in region W and time T. The cluster C is significant if $severity(C) > \delta_s{\cdot}length(T){\cdot}N$, where δ_s is the severity threshold, N is the number of sensors in W and $severity(C) = \sum_{SF} \mu_i = \sum_{TF} \nu_j$.

Note that, the system measures the cluster significance by a relative threshold δ_s, because $severity(C)$ is influenced by the query scales, *e.g.*, the severities of high-level clusters in one month are usually larger than the low-level clusters in a day.

The key challenge for online clustering is to prune the trivial micro-clusters and meanwhile guarantee the accuracy of significant macro-clusters. One strategy is beforehand pruning: the system pushes down the prune step to lower levels by only selecting the significant micro-clusters for integration. However this strategy cannot guarantee finding all the significant macro-clusters, because a micro-cluster that contributes to a significant macro-cluster may not be significant by itself. If the algorithm prunes all insignificant micro-clusters beforehand, the severity of the macro-cluster will also be reduced and may not be significant anymore.

Example 6. The micro-clusters of Los Angeles in Oct. 30th are shown in Figure 11 (a) and the monthly significant macro-clusters A and B are plotted in Figure 11(b). In Figure 11(a), the micro-clusters a, b, j, k and o are going to be integrated as parts of the significant macro-clusters even if they are relatively trivial. The micro-clusters e, h and i are significant in the scale of one day, but actually they can be pruned since they have no contribution for any significant macro-clusters in one month.



(a) Micro-clusters in Oct. 30th

(b) Significant Macro-clusters in October

Fig. 11. Example: Problem of Beforehand Pruning

Can we foretell which micro-cluster will become a part of the significant macro-clusters and which will not? If the system knows such guiding information, it can improve query efficiency and meanwhile guarantee the result's accuracy. The heuristic comes from the bottom-up method: Recall that the bottom-up method uses total severity $F(W, T)$ as the measure.

Property 4. The total severity $F(W, T)$ is a distributive measure.

As a distributive measure, $F(W, T)$ is efficient to compute [9] and can be employed as the guidance to retrieve significant clusters. However, one may worry that $F(W, T)$ is computed on pre-defined regions such as zipcode areas and their boundaries are different from the atypical clusters. Fortunately, Property 5 shows that there is a connection between pre-defined regions and atypical clusters.

Property 5. Given a query $Q(W, T)$ and the relative severity threshold δ_s, for a spatial region $W' \subseteq W$, if $F(W', T) < \delta_s \cdot length(T) \cdot N$, where N is the number of sensors in W, then there is no significant macro-cluster in W' within time T.

Property 5 can be used to help filtering the micro-clusters. The system only needs to integrate the clusters in the regions where the total severities are larger than threshold, *i.e.*, the *red-zones*.

Example 7. In Figure 12, the red-zones are tagged out. They are generated by the bottom-up styled model with a pre-defined zipcode area hierarchy. The micro-clusters e, g, i and m can be pruned safely since they are outside the zones; a, b and d should be kept for clustering since they are in the zones; c, k, f, o and n are also kept since they intersect with the red zones and may contribute to the significant macro-clusters.

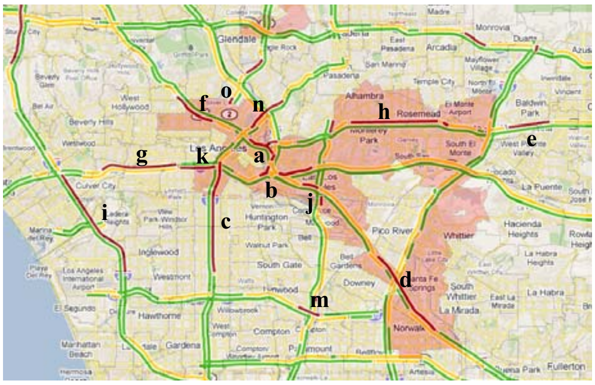


Fig. 12. Red Zone Guided Clustering

Figure 13 shows detailed steps of red-zone guided online clustering: The system first computes the severity on pre-defined regions in bottom-up method and retrieve the red zones (Lines 1), then removes the micro-clusters that locate outside those red zones (Lines 2 – 3). The cluster integration algorithm (Algorithm 3) is called to generate the macro-clusters (Line 4). Since the algorithm can only guarantee there is no false negatives (*i.e.*, not missing any significant macro-clusters), it is possible to generate some false positives. A check procedure is processed to prune the clusters without enough severity at the last step (Lines 5 – 7).

The major cost of Algorithm 4 is at Line 8 to call the clustering algorithm. In the worst case, no cluster could be filtered out and the algorithm's time complexity is still quadratic to the number of micro-clusters. However, in our experiments, about 80% micro-clusters could be filtered out with reasonable δ_s and the query efficiency was improved dramatically.

```
Algorithm 4. Red Zone Guided Clustering
Input: query $Q(W,T)$, severity threshold $\delta_s$, similarity threshold
  $\delta_{sim}$, micro-cluster set micro_set in atypical forest
Output: the significant macro-cluster set sig_set
compute $F(W_i,T)$ in each region $W_i$;
if $F(W_i,T) < \delta_s*length(T)*N$ then
    remove the sensors inside $W_i$ from micro_set;
sig_set ← integration (micro_set, $\delta_{sim}$); //Algorithm 3
for macro-cluster $C$ in sig_set
    if $severity(C) < \delta_s*length(T)*N$ then
        remove $C$ from sig_set;
return sig_set;
```

Fig. 13. Algorithm: Red-zone Guided Clustering

Dataset	Date	Sensor. #	Reading. #	Atypical Data%
D_1	Oct. 2008	4,076	$3.4*10^7$	~2.3%
D_2	Nov. 2008	4,052	$3.3*10^7$	~3.7%
…	…	…	…	…
D_{12}	Sep. 2009	4,076	$3.3*10^7$	~4.0%
the severity threshold δ_s: 2% — 20%, default 5%				
the distance threshold δ_d: 1.5 mile — 24 mile, default 1.5 mile				
the time interval threshold δ_t: 15 min — 80 min, default 15 min				
the similarity threshold δ_{sim}: 0.1 — 1, default 0.5				
the g function: max, min, arithmetic mean, harmonic mean and geometric mean, default: arithmetic mean				

Fig. 14. Experiment Settings and Parameters

V. Performance Evaluation

Since the idea of this study is motivated by application problems, we use real world datasets to evaluate the proposed approaches. Twelve datasets are collected from the PeMS traffic monitoring system [2], each stores one-month traffic data in the areas of Los Angeles and Ventura. The data are collected from over 4,000 sensors on 38 highways. There are more than 1.1 million records for a single day and totally 428 million records for the whole year. The total size of all the datasets is over 54 GB.

The experiments are conducted on a PC with an Intel 2200 Dual CPU @ 2.20G Hz and 2.19G Hz. The RAM is 1.98 GB and the operating system is Windows XP SP2. All the algorithms are implemented in Java on Eclipse 3.3.1 platform with JDK 1.5.0. The detailed experimental settings and parameters are listed in Figure 14.

A. Evaluations of Model Construction

In this subsection we evaluate the algorithms of offline model construction. *CubeView* [15] is a bottom-up method on traffic data. The original CubeView algorithm aggregates all the traffic records with pre-defined spatial and temporal hierarchies. In this experiment, the system carries out a pre-processing step to select atypical records and adjusts CubeView to construct the model only on the atypical data. We first construct the models on a single dataset, then gradually increase the number of datasets, until all the twelve datasets are used in the experiment. Figure 15 shows the time costs of the original Cubeview (OC), modified CubeView (MC), the pre-processing step (PR) and our atypical cluster method (AC). The x-axis is the number of datasets used in the experiment and y-axis is the time cost in logarithmic scale. MC and AC are an order of magnitude faster than OC because they are constructed on the atypical data, which are only 2% to 5% of the original datasets. The time cost of PR is close to OC since both of them have to scan the original datasets with huge I/O overhead. However the pre-processing step only needs to carry out once for constructing different models.

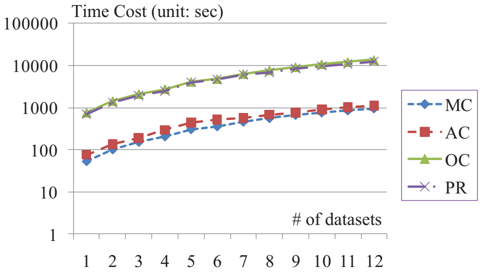


Fig. 15. Efficiency: Construction Time Cost vs. # of Datasets

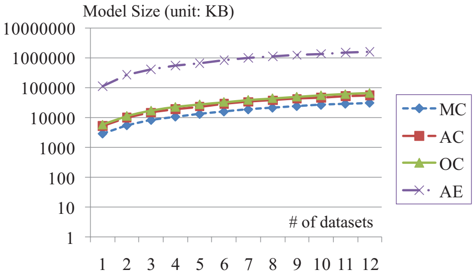


Fig. 16. Size: Constructed Model Size vs. # of Datasets

Figure 16 shows the constructed model size of original Cubeview (OC), modified CubeView (MC) and atypical cluster method (AC). The size of corresponding atypical events (AE) is also recorded in the figure. MC achieves the best compression effects since it only records the numeric measure of total severity, but it cannot describe the complex atypical events. AC stores all the critical information about spatial and temporal features of AE, but only costs 0.5% to 1% space of AE.

B. Comparisons in Analytical Query Processing

In this subsection we evaluate the performances of analytical query processing. Three query processing strategies are compared: (1) integrating all the micro-clusters (All); (2) pruning the insignificant clusters beforehand (Pru); (3) the red-zone guided clustering (Gui).

In the experiments, the system only pre-computes the micro-clusters of each day. The analytical query's spatial range is fixed as Los Angeles and time range gradually increases from one week (requiring to aggregate the micro-clusters of 7 days) to three months (84 days). Figure 17 (a) and (b) record the average time and I/O costs (measured by the number of input

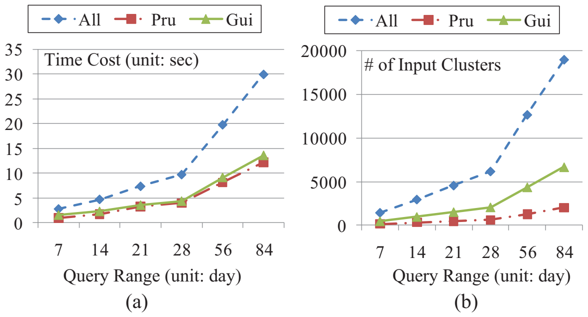


Fig. 17. Efficiency: Query Time and I/O Costs

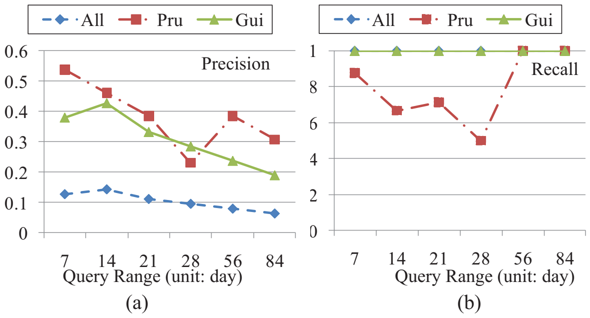


Fig. 18. Effectiveness: Precision and Recall *w.r.t.* Query Range

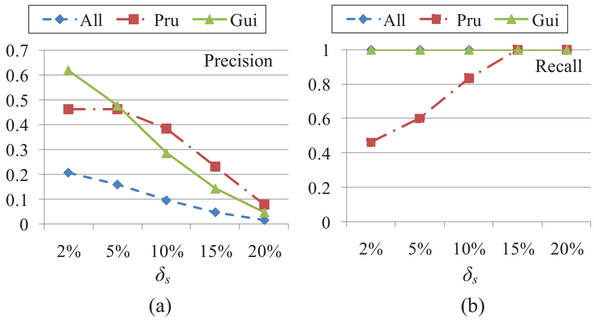


Fig. 19. Effectiveness: Precision and Recall *w.r.t.* δ_s

micro-clusters). Although Gui has extra cost to compute the red-zones, the time efficiency is still close to Pru. From the figure one can see clearly that Gui and Pru are much more efficient than All. Gui's time cost is only about 15%–20% of All.

To evaluate the effectiveness of query results, we compute the precision and recall of the significant clusters. Since the integrating-all method prunes no clusters, its results contain all the significant clusters. The system checks the results of All and retrieves the true significant clusters as the ground truth. The measures of precision and recall are then computed as follows.

Precision: The precision is calculated as the proportion of significant clusters in the returned query results.

Recall: The recall is the proportion of retrieved significant clusters over the ground truth.

The system increases the query time range from 7 days to 84 days and records the precision and recall of three methods in Figure 18. For all the methods, their precision decreases with larger query range, because the cluster severity does not grow linearly *w.r.t.* the query range, and the significant clusters are inevitably fewer in larger query range with fixed severity threshold. In the experiment, Pru has the highest precision, because it prunes all the trivial micro-clusters and generates fewer macro-clusters (Figure 18 (a)). However, as shown in Figure 18 (b): Pru cannot guarantee to find all the significant clusters. Its recall might even be lower than 50% in some cases. Therefore, even Pru is the winner on efficiency and precision, it is not feasible to process analytical query since the significant clusters may be missed in the results.

In the next experiment, we fix the query time range as 14 days and evaluate the influence of severity threshold δ_s. The experimental results are shown in Figure 19. The precision drops with larger δ_s, because fewer macro-clusters can meet the high standard of severity to become significant. Another interesting observation is that, the recall of Pru increases when δ_s grows. Pru is unlikely to miss the macro-clusters with very high severities. However, the detailed spatial and temporal features of those clusters may not be accurate, because Pru filters out some micro-clusters that should be integrated in.

The precision of Gui in the above experiments is not high, however this measure can be easily improved. The system can efficiently filter out the false positives and guarantee 100% precision by checking the macro-cluster's total severity. Gui has such a procedure (Lines 5 – 7 in Algorithm 4). This procedure is turned off in the experiments for a fair play.

C. Parameter Tuning for Atypical Cluster Method

In the last experiment, we study the influence of the parameters in the atypical cluster based method, including time interval threshold δ_t, distance threshold δ_d, similarity threshold δ_{sim} and balance function g. The system first retrieves the micro-clusters in each day with different δ_t and δ_d, then carries out the cluster integration to generate the macro-clusters for every week and month. Figure 20 (a) shows the average number of the atypical clusters in every day, week and month. The figure also records the average number of weekly/monthly significant clusters as sig(week)/sig(month). One can clearly see that, the numbers of weekly and monthly macro-clusters are much larger than the micro-clusters, but most of them are the trivial ones. Only 0.1% to 0.5% of those macro-clusters are significant. When δ_t increases, more clusters can be merged together and the numbers of macro-clusters decrease rapidly. But the numbers of significant macro-clusters are more stable. Since those significant clusters have already integrated a large amount of micro-clusters, they can hardly merge with each other due to large difference on spatial and temporal features. In Figure 20 (b) we record the numbers of atypical clusters with different δ_d. The influence of δ_d is smaller than δ_t. The number of significant cluster is also robust to this parameter.

We also study the influences of similarity threshold δ_{sim}

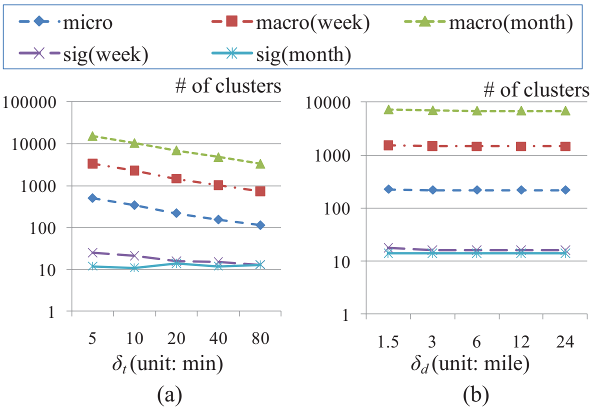


Fig. 20. Size: # of Clusters vs. δ_t and δ_d

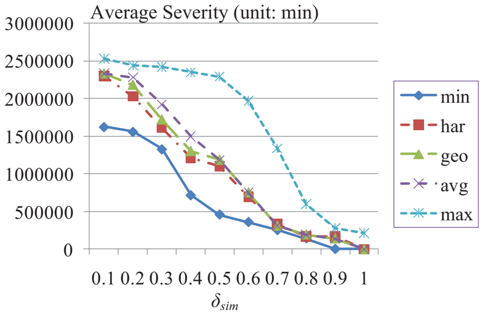


Fig. 21. Average Severity of Significant Cluster vs. δ_{sim}

and the balance function g in Equations (3) and (4). Since they only influence the cluster integration results, we carry out the integration process with various balance functions, including max, min, the arithmetic mean (avg), the geometric mean (geo) and harmony mean (har). Figure 21 shows the average severity of the significant clusters *w.r.t.* δ_{sim}. Generally speaking, the max function integrates more clusters and the min function is the most conservative. From Figures 20 and 21, we can also learn that the result of significant cluster is robust to the time interval threshold δ_t and distance threshold δ_d, but the severity of significant clusters may reduce rapidly with larger similarity threshold δ_{sim}. Hence δ_{sim} should not be set too high, we suggest to set it around 0.5.

D. Discussions

In this study, we illustrate the atypical cluster techniques mainly on spatial-temporal dimensions and carry out the performance evaluation on traffic data, because: (1) the spatial and temporal dimensions are actually the most basic and important dimensions in many CPS applications, and the user's queries are usually related to such two dimensions; (2) Large volume of traffic data is open to public; and (3) The atypical events of traffic (*i.e.*, the congestions) are actually more complex than in many other domains. Apart from spatial and temporal dimensions, the users may require to analyze the data on other domain specific dimensions. For example, in the traffic system, the transportation officer may want to check the congestions related to bad weather or the accident reports. The proposed framework can be easily extended to support the analysis on such context dimensions. The weather dimension can be joined with temporal dimension with the date and the accident dimension can be joined with temporal and spatial dimensions by the accident time and location. By joining those dimension information, the system can support analytical queries on more dimensions.

The clustering and integration methods used in this study are all "hard-clustering", *i.e.*, a data record can only belong to one micro-cluster and a micro-cluster could only be merged to one macro-cluster. Hence it is possible that the clustering result may not be deterministic. However, the influence is limited for the analytical query, because the macro-clusters are usually aggregated from hundreds of micro-clusters and there is almost no difference on merging a single micro-cluster or not.

VI. Related Works

PeMS is a freeway performance monitoring system in California [5]. It collects giga-bytes data each day to produce useful traffic information. *CarWeb* is a platform to collect real-time GPS data from cars [11]. When sufficient information has been collected, the system estimates traffic information such as the average speed of vehicles. Google Traffic is a service based on the Google Maps [1]. It automatically includes real-time traffic flow conditions to the maps of thirty major cities of the United States. In a later released version a traffic model is used to predict the future traffic situation based on historical data.

Most systems do not support analytical queries. Some of them, like Google traffic, provide prediction functions but still do not support analysis on historical data.

The pioneering work on spatial data warehousing is proposed by Stefanovic *et al.* with the concepts of *spatial cube*[16]. In [8], Giannotti *et al.* summarize the ideas of *trajectory cube*. Shekhar *et al.* propose a web based visualization tool for intelligent transportation system called *Cubeview* [15]. This method successfully investigates high performance critical visualization techniques for exploring historical traffic data. A traffic incident detection module is developed based on this work [10]. Ali *et al.* propose a PDT framework with pre-defined phenomena model to track events in a continuous SQL queries [4]. It is a big improvement for event detection and analysis on the data streams. This work is further extended for aggregate location monitoring in sensor network. The proposed methods can provide high-quality location monitoring services [6].

Papadias *et al.* design efficient OLAP operations based on R-tree index [12]. The *aggregation R-tree* defines a hierarchy among MBRs that forms a data cube lattice. In a later study [13], the authors extend the indexing techniques to spatial and temporal dimensions. *Historical RB-tree* is built to help aggregating the measures on static and dynamic regions. The *aggregate point-tree* is proposed to solve range aggregate queries [21]. In [20], Tao *et al.* combine sketches with spatio-temporal aggregate indexes to solve the distinct counting problem.

However, those techniques are not feasible for analyzing the atypical events in CPS data. The main reason is about their measures: most methods employ numeric measures and aggregate them in pre-defined hierarchies. They cannot describe the complex atypical events. In addition, those spatial aggregations must be carried out in pre-defined regions (*e.g.*, R-tree rectangle, zipcode area, etc), but the atypical events may not follow the fixed boundaries.

VII. Conclusions and Future Work

In this paper, we have investigated the problem of multidimensional analysis of atypical events in cyber-physical systems. A new model of atypical cluster is designed to describe the atypical events in CPS data. The red-zone guided clustering algorithm is proposed to efficiently retrieve the significant clusters. Our experiments on large real datasets show the feasibility and scalability of proposed methods.

This paper is our first step in the CPS data analysis. In the future we will extend the atypical event analysis to support more complex applications, such as the event prediction and trustworthiness analysis in atypical data. We are also interested in applying the proposed methods to more applications, such as intruder detection on battlefields.

VIII. Acknowledgment

The authors would like to thank the anonymous reviewers for their insightful and constructive comments.

The work was supported in part by U.S. NSF grants IIS-0905215, CNS-0931975, CCF-0905014, IIS-1017362, the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA). Sangkyum Kim is supported in part by National Science Foundation (award number OCI 07-25070). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] http://maps.google.com.

[2] http://pems.dot.ca.gov.

[3] Supplement to the presidents budget for fiscal year 2008. In *The Networking and Information Technology Research and Development Program*, 2007.

[4] M. Ali, M. F. Mokbel, W. Aref, and I. Kamel. Detection and tracking of discrete phenomena in sensor-network databases. In *SSDBM*, 2005.

[5] T. Choe, A. Skabardonis, and P. Varaiya. Freeway performance measurement system (pems): An operational analysis tool. In *TRB*, 2002.

[6] C.-Y. Chow, M. F. Mokbel, and T. He. Aggregate location monitoring for wireless sensor networks: A histogram-based approach. In *MDM*, 2009.

[7] C.Lu and J. Zheng. Aitvs: Advanced interactive traffic visualization system. In *ICDE*, 2006.

[8] F. Giannotti and D. Pedreschi. *Mobility, Data Mining and Privacy*. Springer, 2008.

[9] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. *Data Mining and Knowledge Discovery*, 1:29–54, 1997.

[10] Y. Jin, J. Dai, and C. Lu. Spatial-temporal data mining in traffic incident detection. In *SDM Workshop: Spatial Data Mining*, 2006.

[11] C. Lo and W. Peng. Carweb: A traffic data collection platform. In *MDM*, 2008.

[12] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao. Efficient olap operations in spatial data warehouses. In *SSTD*, 2001.

[13] D. Papadias, Y. Tao, P. Kalnis, and J. Zhang. Indexing spatio-temporal data warehouses. In *ICDE*, 2002.

[14] I. D. Payne. Eisenhower transportation fellowships. *Public Roads*, 63(2), 1999.

[15] S. Shekhar, C. Lu, R. Liu, and C. Zhou. Cubeview: A system for traffic data visualization. In *ICITS*, 2002.

[16] N. Stefanovic, J. Han, and K. Koperski. Object-based selective materialization for efficient implementation of spatial data cubes. *IEEE TKDE*, 12(6), 2000.

[17] L. Tang, B. Cui, and H. Li. Effective variation management for pseudo periodical streams. In *SIGMOD*, 2007.

[18] L. Tang, X. Yu, S. Kim, J. Han, C. Hung, and W. Peng. Tru-alarm: Trustworthiness analysis of sensor networks in cyber-physical systems. In *ICDM*, 2010.

[19] L.-A. Tang, X. Yu, S. Kim, and J. Han. Tru-alarm: Trustworthiness analysis of sensor networks in cyber-physical systems. In *ICDM*, 2010.

[20] Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias. Spatio-temporal aggregation using sketches. In *ICDE*, 2004.

[21] Y. Tao and D. Papadias. Range aggregate processing in spatial databases. In *IEEE TKDE*, 2004.

[22] X. Yu, L. Tang, and J. Han. Filtering and refinement: A two-stage approach for efficient and effective anomaly detection. In *ICDM*, 2009.

Appendix

Property 1. The atypical event is a holistic model.

Proof: A model is *holistic* if there is no constant upper bound on the storage size needed to describe a sub-aggregation [9].

The atypical event contains the atypical records from CPS dataset. There is no bound for the number of such records. Let us consider an extreme case, suppose there is a heavy snow and the traffic of the entire city is tied up through the whole day. In such case, this event contains all the atypical records of the dataset, and no constant bound of storage size can be found. Hence the model of atypical event is holistic. ■

Property 2. The spatial and temporal features in atypical clusters are algebraic.

Proof: A feature is *algebraic* if it can be computed by an algebraic function with m arguments (m is a bounded positive integer), and each of the arguments is distributive [9]. We will prove that the spatial feature is algebraic in the process of integrating n micro-clusters to a macro-cluster by mathematical induction.

(1) The basis: First we study the case that n=2.

Let S_1 and S_2 be the sensor sets of two micro-clusters, the spatial feature of macro-cluster SF_{macro} is computed as Equation 7.

$$\begin{aligned} SF_{macro} = \{\langle s_i, \mu_i^1 + \mu_i^2 \rangle | s_i \in (S_1 \cap S_2)\} \\ \cup \{\langle s_j, \mu_j \rangle | s_j \notin (S_1 \cap S_2), s_j \in (S_1 \cup S_2)\} \end{aligned} \tag{7}$$

Since the sensor severity μ is distributive, according to the definition, spatial feature is algebraic when n=2.

(2) The inductive step: suppose that the statement holds for $n-1$, we study the case of integrating n micro-clusters.

The macro-cluster C_N can be seen as the integration of the macro-cluster C_{N-1} and the nth micro-cluster C_n. Let S_{N-1} and S_n be the corresponding sensor sets, the spatial feature of macro-cluster SF_N is computed as Equation 8.

$$\begin{aligned} SF_N = \{\langle s_i, \mu_i^1 + \mu_i^2\rangle | s_i \in (S_{N-1} \cap S_n)\} \\ \cup\{\langle s_j, \mu_j\rangle | s_j \notin (S_{N-1} \cap S_n), s_j \in (S_{N-1} \cup S_n)\} \end{aligned} \tag{8}$$

Therefore the statement holds for the case of integrating n micro-clusters. The spatial feature is algebraic.

From the same steps, it is easy to obtain that the temporal feature is also algebraic. ■

Property 3. The operation of merging atypical clusters is mathematically commutative and associative.

Proof: To prove the merge operation is mathematically commutative, we have to show that for any C_1 and C_2, C_1 merge $C_2 = C_2$ merge C_1.

For two clusters C_1 and C_2, the spatial feature of their integrating cluster is SF_{new} computed as

$$SF_{new} = \{\langle s_i, \mu_i^1 + \mu_i^2\rangle | s_i \in (S_1 \cap S_2)\}$$
$$\cup\{\langle s_j, \mu_j\rangle | s_j \notin (S_1 \cap S_2), s_j \in (S_1 \cup S_2)\}$$

The positions of S_1 and S_2 are equal in the above equation, SF_{new} is not influenced by the order of C_1 and C_2. It is the same for temporal feature computation. And the identity feature is generated independently. Therefore for any C_1 and C_2, C_1 merge $C_2 = C_2$ merge C_1. The merge operation is mathematical commutative.

To prove the merge operation is mathematically associative, we have to show that for any C_1, C_2 and C_3, (C_1 merge C_2) merge $C_3 = C_1$ merge (C_2 merge C_3).

Let us denote:
C_4 = C_1 merge C_2; C_5 = C_2 merge C_3;
C_6 = C_4 merge C_3 = (C_1 merge C_2) merge C_3;
C_7 = C_1 merge C_5 = C_1 merge (C_2 merge C_3).

The spatial feature $SF(C_6)$ is computed as:

$$\begin{aligned} SF(C_6) = \{\langle s_i, \mu_i^4 + \mu_i^3\rangle | s_i \in (S_4 \cap S_3)\} \\ \cup\{\langle s_i, \mu_i\rangle | s_i \notin (S_4 \cap S_3), s_i \in (S_4 \cup S_3)\} \end{aligned} \tag{9}$$

Since $S_4 = S_1 \cup S_2$, Equation 9 can be written as:

$$\begin{aligned} SF(C_6) = \{\langle s_i, \mu_i^{1,2} + \mu_i^3\rangle | s_i \in ((S_1 \cup S_2) \cap S_3)\} \\ \cup\{\langle s_i, \mu_i\rangle | s_i \notin ((S_1 \cup S_2) \cap S_3), \\ s_i \in ((S_1 \cup S_2) \cup S_3)\} \\ = \{\langle s_i, \mu_i^1 + \mu_i^2 + \mu_i^3\rangle | s_i \in (S_1 \cap S_2 \cap S_3)\} \\ \cup\{\langle s_i, \mu_i^1 + \mu_i^3\rangle | s_i \in (S_1 \cap S_3), s_i \notin S_2\} \\ \cup\{\langle s_i, \mu_i^2 + \mu_i^3\rangle | s_i \in (S_2 \cap S_3), s_i \notin S_1\} \\ \cup\{\langle s_i, \mu_i^1 + \mu_i^2\rangle | s_i \in (S_1 \cap S_2), s_i \notin S_3\} \\ \cup\{\langle s_i, \mu_i\rangle | s_i \in (S_1 \cup S_2 \cup S_3), \\ s_i \notin (S_1 \cap S_3), s_i \notin (S_2 \cap S_3), s_i \notin (S_1 \cap S_2)\} \end{aligned} \tag{10}$$

Since $S_5 = S_2 \cup S_3$, Equation 10 can be converted to:

$$\begin{aligned} SF(C_6) = \{\langle s_i, \mu_i^1 + \mu_i^{2,3}\rangle | s_i \in (S_1 \cap (S_2 \cup S_3))\} \\ \cup\{\langle s_i, \mu_i\rangle | s_i \notin (S_1 \cap (S_2 \cup S_3)), \\ s_i \in (S_1 \cup (S_2 \cup S_3))\} \\ = \{\langle s_i, \mu_i^1 + \mu_i^5\rangle | s_i \in (S_1 \cap S_5)\} \\ \cup\{\langle s_i, \mu_i\rangle | s_i \notin (S_1 \cap S_5), s_i \in (S_1 \cup S_5)\} \\ = SF(C7) \end{aligned} \tag{11}$$

Equation 11 shows that the spatial features are the same for the macro-clusters C_6 and C_7, so are the temporal features. And the identity feature is generated independently. Hence the merge operation is mathematical associative. ■

Property 4. The total severity $F(W,T)$ is a distributive measure.

Proof: A measure is *distributive* if it can be derived from the aggregation values of n subsets, and the measure is the same as that derived from the entire data set [9].

Let us partition the dataset in region W and time T into n subsets, each with region $W_i \subset W$ and $T_i \subset T$. The severity of the ith subset is computed by aggregating the severities of every subset as shown in following:

$$F(W_i, T_i) = \sum_{s \in W_i} \sum_{t \in T_i} f(s,t)$$

Then total severity is computed as $F(W,T) = \sum_1^n F(W_i, T_i)$.

Since $\bigcup\limits_{i=1,\ldots,n} W_i = W$ and $\bigcup\limits_{i=1,\ldots,n} T_i = T$;

$$F(W,T) = \sum_{s \in W} \sum_{t \in T} f(s,t)$$

Therefore $F(W,T)$ is in the same format from the one derived from the entire dataset, and it is a distributive measure. ■

Property 5. Given a query $Q(W,T)$ and the relative severity threshold δ_s, for a spatial region $W' \subseteq W$, if $F(W',T) < \delta_s{\cdot}Length(T){\cdot}N$, where N is the number of sensors in W, then there is no significant macro-cluster in W' within time T.

Proof. We will prove the statement by contradiction.

Suppose there is a cluster C_j with sensor set $S_j \in W'$ and time window sequence $T_j \subseteq T$, such that $severity(C_j) \geq \delta_s{\cdot}Length(T){\cdot}N$.

Since $F(W',T)$ is the aggregation of total severity in W' and T;

$$F(W',T) \geq severity(C_j) \geq \delta_s \cdot Length(T) \cdot N$$

We now have a contradiction with the condition that $F(W',T) < \delta_s{\cdot}Length(T){\cdot}N$. Hence there does not exist such cluster C_j. ■