Short and Long-term Pattern Discovery Over Large-Scale Geo-Spatiotemporal Data

Sobhan Moosavi, Mohammad Hossein Samavatian, Arnab Nandi, Srinivasan Parthasarathy, and Rajiv Ramnath
Department of Computer Science and Engineering
The Ohio State University
Columbus, Ohio 43210-1277
{moosavi.3,samavatian.1,nandi.9,parthasarathy.2,ramnath.6}@osu.edu

ABSTRACT

Pattern discovery in geo-spatiotemporal data (such as traffic and weather data) is about finding patterns of collocation, co-occurrence, cascading, or cause and effect between geospatial entities. Using simplistic definitions of spatiotemporal neighborhood (a common characteristic of the existing general-purpose frameworks) is not semantically representative of geo-spatiotemporal data. We therefore introduce a new geo-spatiotemporal pattern discovery framework which defines a semantically correct definition of neighborhood; and then provides two capabilities, one to explore *propagation* patterns and the other to explore *influential* patterns. Propagation patterns reveal common cascading forms of geospatial entities in a region. Influential patterns demonstrate the impact of temporally long-term geospatial entities on their neighborhood. We apply this framework on a large dataset of *traffic and weather* data at countrywide scale, collected for the contiguous United States over two years. Our important findings include the identification of 90 common propagation patterns of traffic and weather entities (e.g., *rain* → *accident* → *congestion*), which results in identification of four categories of states within the US; and interesting influential patterns with respect to the "location", "duration", and "type" of long-term entities (e.g., *a major construction* → *more traffic incidents*). These patterns and the categorization of the states provide useful insights on the driving habits and infrastructure characteristics of different regions in the US, and could be of significant value for applications such as urban planning and personalized insurance.

CCS CONCEPTS

• **Applied computing** → **Transportation**; • **Information systems** → **Traffic analysis**; *Information integration*; *Data cleaning*.

KEYWORDS

Propagation Patterns, Influential Patterns, Geo-Spatiotemporal Data

ACM Reference Format:
Sobhan Moosavi, Mohammad Hossein Samavatian, Arnab Nandi, Srinivasan Parthasarathy, and Rajiv Ramnath. 2019. Short and Long-term Pattern Discovery Over Large-Scale Geo-Spatiotemporal Data. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330755

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330755

1 INTRODUCTION

Spatiotemporal pattern discovery has seen considerable interest over the past decade, with various frameworks were proposed to process the data to find interesting patterns [3, 4, 14, 19–21, 24, 30, 33, 34]. The application domains of relevance include public safety, transportation, earth science, epidemiology, climatology, and environmental management [25]. These frameworks can be used to discover patterns of collocation and co-occurrence, interactions and correlations, cascading, sequential, or cause and effect relationship patterns. However, they all rely on a simplistic definition of spatiotemporal neighborhood, essentially spatial closeness based on an Euclidean or Cartesian system and temporal overlap [4, 14, 21, 33], which often makes their use impractical for applications such as traffic, transportation, or weather analyses. For example, a traffic accident on one lane of a freeway has no impact on traffic flow on an opposite lane, yet general-purpose frameworks will locate both lanes in a single neighborhood. Another example arises when studying the impact of a snow event (on traffic flow) which continues well past when the snow event has ended. The time overlap constraint required by existing frameworks would hinder such a study. Note that there may not be any trivial changes to be made to make the existing frameworks semantically applicable for this type of data. Because, their basis is on a specific way of defining spatiotemporal neighborhood, which changing that would make them unusable (e.g., regarding their pruning step) or expensive to be employed.

To address these challenges, we propose a new framework for finding patterns in geo-spatiotemporal data. This framework consists of two parts, one to explore *propagation* patterns, and the other to reveal *influential* patterns. Identifying propagation patterns requires the exploration of partially ordered sets of geospatial entities, that are spatially co-located and temporally co-occurring, with potential "cause and effect" relationships between the entities. An example of this type is a `rain` event, which causes an `accident`, with the accident then causing `congestion`. Identifying influential patterns, on the other hand, requires studying the impact of temporally long-term geospatial entities (e.g. a major construction) on their spatial neighborhoods. An example of this type of pattern is the increase in number of `congestion` events in a region because of a long-term `snowing` event.

To explore propagation patterns – also referred as "cascading patterns"[21] or "spatiotemporal couplings" [25], we propose a tree-pattern-mining-based process, we term *short-term pattern discovery*, which employs a strict definition of spatial neighborhood to ensure spatial collocation, and a definition of temporal co-occurrence specific to geo-spatiotemporal data and application domain constraints. To explore influential patterns – also referred as "tele-couplings" [25] – we propose a new process, we term *long-term pattern discovery*, to examine the effect of long-term entities on their neighborhood to reveal any significant impact. As in, and drawing from [11, 16], this process may be used to study impacts with respect to different *types*, different *locations*, and *duration* of long-term geospatial entities.

To evaluate our framework, we used a large-scale, real-world geo-spatiotemporal dataset of traffic and weather data. This dataset

covers the contiguous United States[1], includes data collected from August 2016 to August 2018, and contains about 13.1 million instances of traffic entities (e.g., accident, congestion, and construction), and about 2.2 million instances of weather entities (e.g., rain, snow, and storm). Through the processes mentioned above, we found 90 common patterns of propagation of relatively short-term traffic or weather entities, and identified *four* categories of states based on these patterns. In addition, we carefully studied the impact of relatively long-term traffic or weather entities on traffic, and identified a variety of insights with respect to "location", "type", and "duration" of the entities. The main contributions of this paper are as follows:

- Short-term pattern discovery: We propose a new process for discovering propagation patterns in geo-spatiotemporal data, which models spatiotemporal collocation and co-occurrence in terms of tree structures, and adopts an existing tree pattern mining approach to reveal prevalent patterns. In comparison to the general purpose frameworks, this method better suits application domain requirements of a stricter definition of spatiotemporal neighborhood.
- Long-term pattern discovery: We propose a new process for discovering influential patterns in geo-spatiotemporal data, which examines the impact of long-term geospatial entities on their neighborhood in order to reveal significant influential patterns. Exploring such patterns with existing frameworks is not feasible, due to lack of effective spatiotemporal neighborhood metrics to explore longer-term (or lagging) impacts.
- Data collection and processing: We present a set of processes for collecting real-time traffic and historic weather data, using which we built a publicly available "research dataset" of 13.1 million traffic entities (e.g., accident, congestion, and construction), and 2.2 million weather entities (e.g., rain, snow, and storm). This dataset is accessible from https://smoosavi.org/datasets/lstw.
- Findings and insights: By applying our new framework on the above dataset, we present a range of insights for different regions in the United States. These insights may be further utilized for applications such as urban planning, exploring flaws in transportation infrastructure design, traffic control and prediction, impact prediction, personalized insurance, potentially with relevance to the creation of smart cities.

The rest of this paper is organized as follows: We review the related work in Section 2, and provide preliminaries in Section 3. Section 4 describes the dataset preparation, followed by description of framework in Section 5. Experiments and results are presented in section 6, and Section 7 concludes the paper.

2 RELATED WORK

Spatiotemporal pattern discovery has been thoroughly discussed in literature [3, 4, 14, 19–21, 24]. Earlier work focused more on spatial prevalence and paid less attention to temporal aspects [14], while later work considered both aspects simultaneously [25]. The common process of spatiotemporal pattern discovery is to first define spatiotemporal co-occurrence and collocation criteria; then introduce an interest measure (e.g., participation index); and finally outline a *miner* algorithm to find interesting patterns [14]. Techniques in these papers being general purpose solutions, rely on simplistic definitions of collocation (spatial) and co-occurrence (temporal), and unable to reveal complex spatiotemporal correlations (such as influential patterns). Further, they have been developed and only tested on small-scale (real-world or synthetic) data. To address these challenges with respect to geo-spatiotemporal data, we propose a new framework which provides an appropriate and precise definition of collocation and co-occurrence criteria. Moreover, we outline the process of finding complex spatiotemporal patterns and prove its applicability through extensive experiments. Lastly, we apply our framework on a large-scale, countrywide geo-spatiotemporal dataset of traffic and weather data to explore interesting patterns.

Regarding the application domain, there are numerous studies for finding patterns in traffic and weather data, with the following goals: to study the impact of precipitation on likelihood or severity of accidents [7, 16, 28]; to explore the impact of weather on traffic intensity [5, 31]; to reveal the effect of climate change and weather condition on road safety [1, 11, 29]; to characterize road accidents locations [18]; or, to discover frequent spatiotemporal patterns in traffic data [15, 17, 19]. The scale of data in most of these studies is limited to one or at most a few cities. Moreover, interactions and correlations between the different types of traffic entities (accident, congestion, etc.) has not been studied before. Although similar ideas to explore long-term patterns have been previously suggested [7, 11, 16], we extend them by: 1) examining a wider range of weather and traffic entity types besides precipitation; 2) exploring properties of different "locations"; and 3) analyzing the impact of "duration length" on traffic flow.

3 PRELIMINARIES AND PROBLEM

In this section, we first provide preliminaries and definitions, and then present the problem statement. Note that some of the definitions are customized for our illustration application domain (i.e., traffic and weather data). However, this will not limit their generalizability to the other related domains.

3.1 Definitions

- Geospatial Entity: a geospatial entity e is represented by a tuple $\langle type, start, end, loc\rangle$, which shows an entity of type $type$, happened in time interval $[start, end]$, and its location is specified by loc. Definition of loc is related to the application domain. For traffic data, we have $loc = \langle latitude, longitude, Street\_Name, Street\_Side, Zipcode, City, State\rangle$, where $Street\_Side$ shows the relative side of a street (i.e., R or L). For weather data, we have $loc = \langle airport\_code\rangle$, which represents the "airport" that e is reported from its weather station. A geospatial entity is called *long*, if it takes place over a relatively long time interval (see Section 5.2).
- Weak-Dependency Relationship: two *co-occurring* and *co-located* geospatial entities are called weakly dependent. Co-occurrence for two entities e_1 and e_2 means $0 \leq |e_1.start - e_2.start| \leq$ *T-thresh*, where *T-thresh* is a time-threshold. Collocation for two traffic entities requires *location matching* as well as *spatial closeness*. The former means that all location fields except the GPS coordinates should be the same. By latter, we require that $dist(e_1, e_2) \leq$ *D-thresh*, where *dist* is the Haversine distance function [13] based on GPS coordinates, and *D-Thresh* is a distance threshold. With respect to matching a pair of weather and traffic entities, collocation means a match between the "airport station" at which the weather entity is reported and the "airport station" closest to the traffic entity's location.
- Child-Parent Relationship: for two weakly dependent geospatial entities e_1 and e_2, e_1 is a parent for e_2 if e_1 begins before e_2. We treat parent-child relationship as indicative of a *cause and effect* relation. A weather entity may only be the parent (or cause) of a traffic entity, and we do not define such a relationship between two weather entities.
- Tree Structure: given a set of vertices $\mathcal{V} = (v_1, v_2, \ldots, v_n)$, we define tree $T = (V, E)$, where $V \subset \mathcal{V}$ and $E = \{e_1, e_2, \ldots, e_m\}$ is a set of edges, and each edge $e \in E$ connects a pair of vertices $v_i, v_j \in V$ using an *un-directed* edge. A tree is an *acyclic* graph, and vertices with the same parent are *siblings*. Trees in this work have a *root* node, sibling nodes are *un-ordered*, and nodes are *labeled*. Figure 1-(a) shows several examples of such tree structure. In this work, each node of a tree is a geospatial entity, and each edge shows a child-parent relationship between two entities.

[1]The contiguous United States excludes Alaska and Hawaii, and considers District of Columbia (DC) as a separate state.

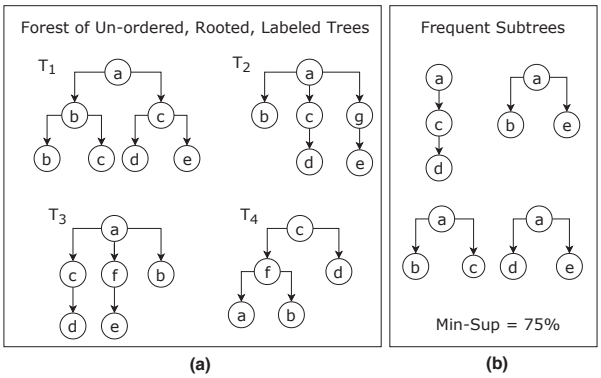


Figure 1: (a) A forest of four trees, (b) Four of embedded frequently occurred subtrees with a minimum support 75%.

- Embedded Subtree: given a tree $T = (V, E)$, we define a subtree as $S = (V', E')$, where $V' \subset V$ and $E' \subset E$. A subtree S is said to be an *embedded subtree* of T if for each edge $e = (v_a, v_b) \in E'$, v_a is an ancestor (and not necessarily the parent) of v_b in T.

3.2 Short and Long-term Pattern Discovery

We now formalize the two related problems studied in this paper.

3.2.1 Short-term Pattern Discovery. Here we seek to find common short-term *propagation patterns* that indicate *how* geospatial entities cause other entities to happen. We represent a set of weakly dependent geospatial entities as un-ordered, rooted, labeled trees, where the entities are nodes, weak dependency relations are the edges, and entity types (e.g., rain, accident, and congestion) are the labels of the nodes. Thus, given a forest $F = \{T_1, T_2, \ldots, T_k\}$ of such trees, the short-term pattern discovery problem is about finding all embedded subtrees in F which are occurred relatively frequently. Formally, for a subtree S and tree T we define $support(S, T)$ by Equation 1:

$$support(S, T) = \begin{cases} 1 & \text{if } S \text{ is a subtree of } T \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Then, we define $support(S, F)$ by Equation 2:

$$support(S, F) = \frac{\sum_{T \in F} support(S, T)}{|F|} \quad (2)$$

For a subtree S, if $support(S, F) \geq min\_sup$, where min_sup is a minimum support threshold, then we say S is a frequent embedded subtree in F. An example of a forest with some of frequently occurring subtree patterns is shown in Figure 1. In this example, we have a forest which includes four trees. Using a minimum support threshold of 75%, we identified several frequently occurring embedded sub-tree patterns, four of which are shown in Figure 1-(b). We use "short-term pattern discovery" to indicate that we search for patterns of immediate or short-term impacts, as opposed to long-term impacts which is discussed next.

3.2.2 Long-term Pattern Discovery. Long-term pattern discovery is about exploring the *magnitude of impact* of long-term geospatial entities on their neighborhood. As an example, consider a *major construction* event in region A, because of which, we might observe more congestion events in the same region (when compared to a time when there was not such a construction event). Given a long entity L, let $\mathcal{S}_R = [e_1, e_2, \ldots]$ be the set of geospatial entities in the *vicinity* of that, where R is the maximum distance threshold[2]. Let $L.start < e.start$ and $L.end > e.end$, $\forall e \in \mathcal{S}_R$. To study the impact of a long entity, we also define two other sets, $\mathcal{S}$*–before*$_R$ and $\mathcal{S}$*–after*$_R$. The former contains all geospatial entities which happened within distance R from L, during a time interval of the same length as L, but before L started. The latter contains all entities in the same neighborhood as L, during a time interval of the same length as L, but which happened after L ended. Given sets $\mathcal{S}_R$, $\mathcal{S}$*–before*$_R$, and $\mathcal{S}$*–after*$_R$, we define the problem of "long-term pattern discovery" as exploring any significant difference between size of set $\mathcal{S}_R$ and the other two sets. In other words, a statistically significant difference between the number of entities when a long entity like L is present, and the number of entities before or after L, shows the magnitude of the impact. We call such an occurrence a long-term or influential pattern.

3.2.3 Connection Between Problems. Short-term pattern discovery is about finding *immediate* impacts, and long-term pattern discovery is about exploring the "long-lastingness" of impacts (i.e., *lagging* impacts). Hence, these two are *complementary* problems, with each one focused on a separate aspect of dependency and pattern discovery, while using the same set of input data.

[2]For each $e_i \in \mathcal{S}_R$, its location is within distance R from L.

4 DATASET

In this section, we describe the dataset preparation process. The resulting dataset includes 13.1 million traffic and 2.2 million weather entities, which are collected from August 2016 to August 2018. The dataset is available at https://smoosavi.org/datasets/lstw.

4.1 Traffic Data

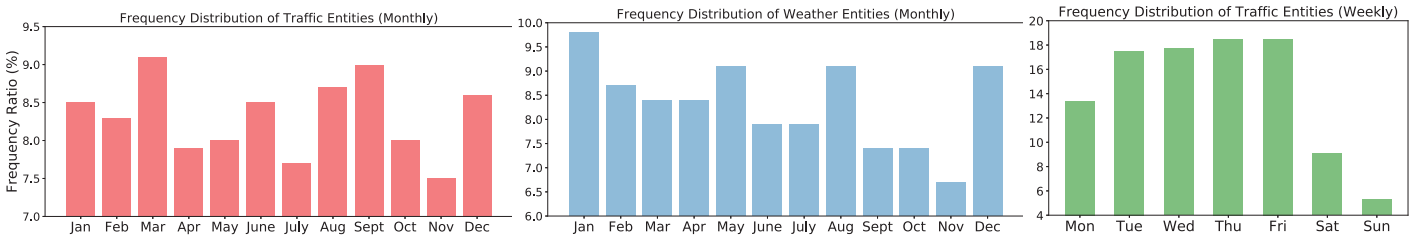
4.1.1 Data Collection Process. To begin with, traffic entities were collected in real-time using a rest API provided by *MapQuest* [32] for a period of two years, from August 2016 to August 2018. To our knowledge, this API broadcast traffic entities captured by a variety of mechanisms - the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road-networks. Traffic data was collected for the contiguous United States (49 States). As the raw traffic entities came with GPS coordinates, we employed *Nominatim* tool [22] to perform reverse geocoding and translated GPS coordinates to addresses.

4.1.2 Data Cleaning Process. Following cleaning steps are employed:

i. Resolving duplicates: Duplicates were identified either explicitly by id (i.e., two entities have the same id), or implicitly by content (i.e., two entities of the same type occurring at the same time and location). We kept one entity and removed the other.
ii. Denoising the data: In this context, noise is related to the "type" of entity, where the Traffic Message Channel (TMC) [8] code (as part of the information for each traffic entity) was different from the default type reported by the MapQuest API. In order to deal with this mismatch, we first extracted 250 different TMC codes from our data, and manually created a new taxonomy by defining a *unified type* for each TMC code using [8] as reference. Finally, we replaced the new taxonomy with the default one in traffic data.

4.1.3 Data Entity Description. We defined the following taxonomy for traffic entities:

- Accident: a common type, which may involve one or more vehicles, and could result in fatality.
- Broken-Vehicle: refers to the situation when there is one (or more) disabled vehicle(s) in a road.
- Congestion: refers to the situation when the speed of traffic is slower than the expected speed. Using the TMC codes, we defined severity of a congestion as *light*, *moderate*, or *heavy*.
- Construction: an on-going construction or maintenance project on a road.
- Event: situations such as *sports event*, *concerts*, or *demonstrations*, that could potentially impact traffic flow.
- Lane-blocked: refers to the cases when we have blocked lane(s) due to traffic or weather condition.



(a) Monthly Traffic Distribution (b) Monthly Weather Distribution (c) Weekly Traffic Distribution

Figure 2: Relative frequency distribution of traffic and weather data, collected from Aug 2016 to Aug 2018, for the contiguous United States.

Table 1: Details on Traffic Dataset, collected for the contiguous United States from Aug 2016 to Aug 2018.

Entity Type	Raw Count	Relative Frequency
Accident	1,169,507	8.9%
Broken-Vehicle	308,112	2.34%
Congestion	10,542,020	80.18%
Construction	209,933	1.60%
Event	32,817	0.25%
Lane-Blocked	246,832	1.88%
Flow-Incident	637,489	4.85%
Total	13,146,710	100%

- Flow-incident: refers to all other types of traffic entities. Examples are *broken traffic light* and *animal in the road*.

Table 1 provides more details on the traffic dataset. The most frequent entity type is "congestion" which includes about 80% of the data, and "accident" is the second most frequent entity type. Figure 2a also depicts the monthly frequency distribution, where the most entities are observed in March and September and the least in November. Additionally, *weekly* frequency distribution of traffic entities is shown by Figure 2c, where 'Friday" and "Sunday" are found to be the days with the most and the least number of recorded entities, respectively.

4.2 Weather Data

4.2.1 Data Collection Process. Raw weather data was collected from 1,973 weather stations located in airports all around the country. The raw data comes in the form of observation records, where each record consists of several attributes such as *temperature*, *humidity*, *wind speed*, *pressure*, *precipitation* (in millimeters), and *condition*[3]. For each weather station, we receive several observation records per day, which are recorded upon any significant change in any of the measured attributes.

4.2.2 Threshold Definition Process. To define the taxonomy of weather entities, we require to extract some threshold values. To do so, we used the United State observations of temperature, wind speed, and precipitation amount for rain and snow for a period of *seven* years, from January 2010 to January 2016, and applied K-Means clustering algorithm [12] on each of these attributes. The obtained cluster centers are used as threshold for these attributes. For temperature, we identified five cluster center values (degrees are in Celsius): $-23.7°$, $-8.6°$, $6.7°$, $21.3°$, and $35.8°$; which we refer them as *severe-cold*, *cold*, *cool*, *warm*, and *hot*, respectively. For wind speed, we found three cluster centers $13.2kmh$, $36.2kmh$, and $60kmh$, which we refer them as *calm*, *moderate*, and *storm* windy conditions, respectively. For rain, we identified three cluster centers 2.5, 7.1, and 11.6 millimeters, which we refer them as *light*, *moderate*, and *heavy* rainy conditions, respectively. Lastly, for snow we found three cluster centers 0.6, 1.7, and 2.5 millimeters, which we refer them as *light*, *moderate*, and *heavy* snowy conditions, respectively.

4.2.3 Entity Extraction Process. Given the above threshold values and the raw weather data records from August 2016 to August 2018, we processed each record to use it (if it represents an entity), merge it (if it is part of a previously found entity), or remove it (if it does not represent any entity), and defined the following taxonomy:

- Severe-Cold: extremely cold condition, with $temperature \leq -23.7°$.
- Fog: low visibility condition as a result of *fog* or *haze*.
- Hail: solid precipitation including *ice pellets* and *hail*.
- Rain: rain of any type, ranging from *light* to *heavy*.
- Snow: snow of any type, ranging from *light* to *heavy*.
- Storm: the extremely windy condition, where the wind speed is at least $60kmh$.
- Precipitation: any kind of solid or liquid deposit, but different from snow or rain. This was a generic label we frequently observed in raw weather data.

We extracted 2,178,949 weather entities for a period of two years. Table 2 provides more details on weather data, where the most frequent entity types are "rain", "fog", and "snow". Figure 2b also shows the frequency distribution of weather entities by month; note that most of the entities occurred in January and the least in November.

Table 2: Details on Weather Dataset, collected for the contiguous United States from Aug 2016 to Aug 2018.

Entity Type	Raw Count	Relative Frequency
Severe-Cold	67,285	3.09%
Fog	454,704	20.87%
Hail	1,252	0.06%
Rain	1,384,588	63.54%
Snow	236,546	10.86%
Storm	14,863	0.68%
Precipitation	19,711	0.9%
Total	2,178,949	100%

5 PATTERN DISCOVERY FRAMEWORK

In this section we describe the pattern discovery framework, which consists of two major parts, one for discovery of propagation patterns and the other for influential patterns[4].

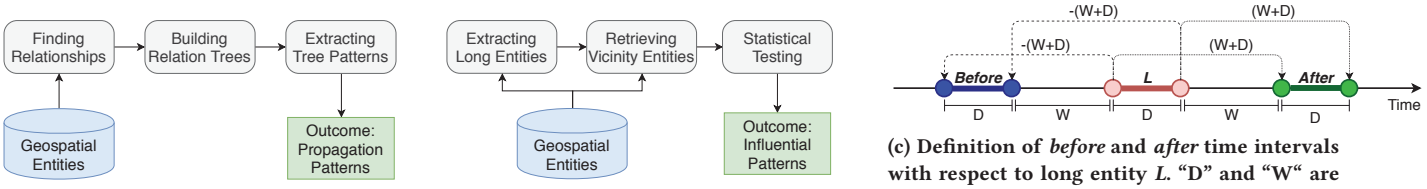
5.1 Short-Term Pattern Discovery

We employed a multi-step process to discover short-term (propagation) patterns in geo-spatiotemporal data. Figure 3a illustrates the process, which includes: 1) finding *child-parent* relationships; 2) building *relation trees*, and 3) extracting *frequent tree patterns*.

- **Finding Child-Parent Relationships**: The first step is to extract all the weakly dependent pairs of entities to define the child-parent relationship for each pair, using the definitions in Section 3.

[3]Possible values are *clear*, *snow*, *rain*, *fog*, *hail*, and *thunderstorm*.

[4]All the implementations in Python are available on GitHub: https://github.com/sobhan-moosavi/ShortLongTerm.



(a) Short-term pattern discovery (b) Long-term pattern discovery (c) Definition of *before* and *after* time intervals with respect to long entity *L*. "D" and "W" are short for Day and Week, respectively

Figure 3: Pattern discovery processes for geo-spatiotemporal data (a) and (b); Defining "before" and "after" time intervals (c).

- **Building Relation Trees**: The next step is to create relation trees from the extracted child-parent relations. Here, tree is a rooted, labeled, un-ordered tree (see Section 3). This step results in a forest of relation trees.
- **Extracting Frequent Tree Patterns**: The last step is to perform frequent tree pattern mining. As described in Section 3, the goal is to extract all frequently observed un-ordered subtrees in our database of relation trees. Examples of such tree patterns with minimum support 75% are shown in Figure 1-b. Here we adopt the SLEUTH algorithm, a growth-based approach proposed by Zaki [35], to extract frequent, embedded, un-ordered sub-trees in our database of relation trees.

5.2 Long-Term Pattern Discovery

In this section we describe the process of long pattern discovery to study the magnitude of impact of long-term entities. This process (shown in Figure 3b) consists of three steps; 1) extracting long-term entities, 2) retrieving vicinity entities, and 3) performing statistical significance testing to explore influential patterns.

- **Extracting Long Entities**: A long (or long-term) entity is one that last for a long time interval, defined by a heuristic threshold. To define such threshold, we first obtain the distribution of duration of entities over the input dataset; and then consider the 99^{th} percentile of the distribution as the threshold which defines long entities. Next, we resolve time and spatial *overlaps* between long geospatial entities using Algorithm 1, to identify and merge overlaps. In this algorithm, we first identify all the conflicted cases for an entity l (lines 2–7); then merge the conflicted entities by updating time, location, and type of l (lines 8–11); and finally we update the list of long entities (line 12). Function *co-occurrence*(.) checks the time-overlap between two entities, and function *collocation*(.) checks the geographical collocation, using distance threshold ρ.
- **Retrieving Vicinity Entities**: After extracting long entities and resolving overlaps, we retrieve entities in the vicinity of each long entity. Thus, given a long entity L, we need to find subsets $\mathcal{S}_R$, $\mathcal{S}$–$before_R$, and $\mathcal{S}$–$after_R$ as follows (R is a maximum vicinity distance):
  - $\mathcal{S}_R$: for this set we look for all those geospatial entities which happened within a distance R from L, with start time strictly after the start time of L, and finished before the end time of L.
  - $\mathcal{S}$–$before_R$: this set is similar to the previous one, except we pick a different time interval to define vicinity, as shown in Figure 3c. Based on this process, we move start and end time of L to $W + D$ days before, where W stands for one week, and D shows duration of L in days. In such an interval, we extract all the entities which happened in vicinity distance R from L.
  - $\mathcal{S}$–$after_R$: similar to the previous one, except we move the start and end time of L to $W + D$ days after.
- **Mining Patterns by Statistical Testing**: Given the set of long entities, we first categorize them into disjoint buckets based on a common characteristic or criteria (e.g., their location or their type). Then, for each bucket, we compare the values of $\mathcal{S}_R$, $\mathcal{S}$–$before_R$, and $\mathcal{S}$–$after_R$ for all long entities, to determine whether there is any significant difference, therefore impact. For this purpose, we design six different testing scenarios and use *two-sample t-test* to test the difference between sample means. For a bucket B, we first calculate the following mean values: μ_L, μ_{before}, and μ_{after} as average of $\mathcal{S}_R$, $\mathcal{S}$–$before_R$, $\mathcal{S}$–$after_R$, respectively, based of the long entities in bucket B. Further, we take the average of $\mathcal{S}$–$before_R$ and $\mathcal{S}$–$after_R$ for each long entity in bucket B, and take the average of average values and denote that by μ_{avg}. Now, we define the following tests for bucket B:
  - T_1: $\mu_{avg} = \mu_L$ versus $\mu_{avg} < \mu_L$. A one-sided test which examines whether the number of geospatial entities during a long entity is larger than this number when there is not such a long entity.
  - T_2: $\mu_{avg} = \mu_L$ versus $\mu_{avg} > \mu_L$. Similar to the previous one, but with the opposite alternative hypothesis.
  - T_3: $\mu_{before} = \mu_L$ versus $\mu_{before} < \mu_L$. A one-sided test which examines whether the number of geospatial entities during a long entity is larger than when the long entity is not started yet.
  - T_4: $\mu_{before} = \mu_L$ versus $\mu_{before} > \mu_L$. Similar to the previous one but with the opposite alternative hypothesis.
  - T_5: $\mu_{after} = \mu_L$ versus $\mu_{after} < \mu_L$. A one-sided test which examines whether the number of geospatial entities during a long entity is larger than when the long entity is ended.
  - T_6: $\mu_{after} = \mu_L$ versus $\mu_{after} > \mu_L$. Similar to the previous one but with the opposite alternative hypothesis.

Note that in all of the above tests, the first condition is the *null hypothesis* and the second one is the *alternative hypothesis*.

Algorithm 1: Merge Geospatial Overlaps

Input: Long entity set $\mathcal{L}$, and distance threshold ρ

```
1  for l in L do
2      List = []
3      for l' in L do
4          if co-occurrence(l, l') and collocation(l, l', ρ) then
5              List.add(l')
6          end
7      end
8      l.StartTime = min_{∀e∈List} StartTime(e)
9      l.EndTime = max_{∀e∈List} EndTime(e)
10     l.location = center_{∀e∈List}(List)
11     l.Type = concat_{∀e∈List}(e.Type)
12     L = L − List
13 end
```

Output: $\mathcal{L}$

6 EXPERIMENTS AND RESULTS

In this section, we describe how the proposed framework was employed to perform pattern discovery. We start with the short-term pattern discovery, and then describe the results for the long-term pattern discovery.

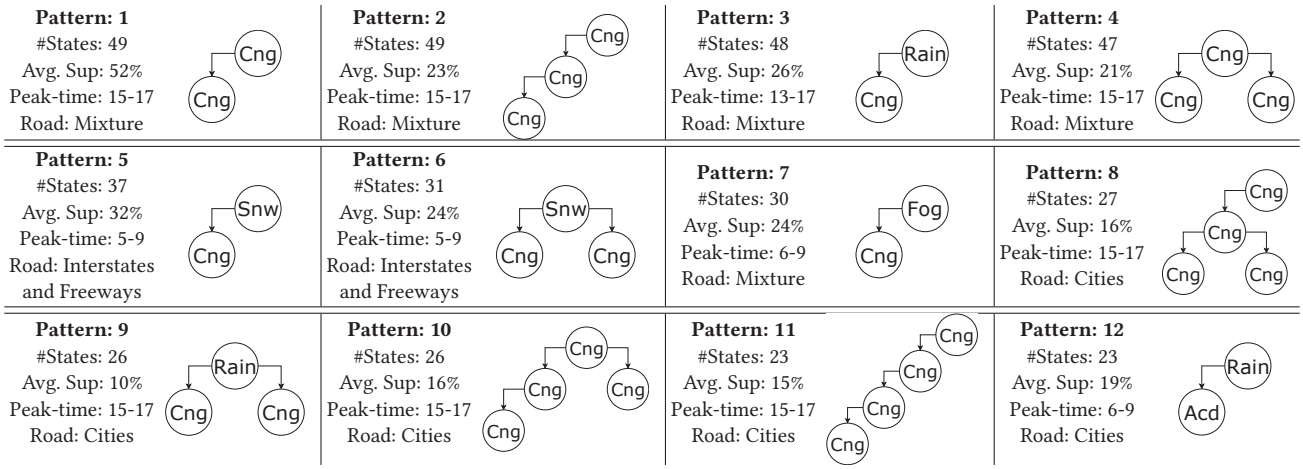


Figure 4: Top frequent embedded tree patterns found based on the short-term dependency relation trees. "Cng", "Acd", and "Snw" are short for congestion, accident, and snow. These patterns show the propagation of traffic/weather entities on a short-term basis.

6.1 Short-term Pattern Discovery Results

First, we extracted all short-term child-parent relationships using thresholds $D\text{-}thresh = 300$ *meters* and $T\text{-}thresh = 10$ *minutes* (see Section 3)[5]. These thresholds were found empirically, with $D\text{-}thresh$ ensuring spatial closeness, and $T\text{-}thresh$ is large enough to consider the delay in a "cause and effect" type of relationship, with respect to our application domain. Using these settings, we found 5, 952, 729 Child-Parent relationships from 15, 325, 659 traffic and weather entities. In total, 39.33% of the traffic entities were found to have at least one weakly dependent weather or traffic entity, and 12.82% of weather entities had at least one weakly dependent traffic entity. Next, we created 1, 723, 637 trees out of 5, 952, 729 child-parent relations. The maximum number of nodes in a tree was found to be 25. Where a traffic entity t had more than one parent, we randomly picked one of them. Given the size of the data and the number of trees, we do not believe that any existing frequent pattern would be missed by this choice. Finally, we employed SLEUTH (Zaki [35]) to extract frequent tree patterns at the *city-level*. More scalable alternatives [26, 27] were an option but not required for our purpose. After extracting frequent patterns for a city, we used these patterns as *core* frequent patterns of the corresponding state. This allows us to account for the potential diversity among different cities in a state (i.e., based on population, traffic, and/or weather condition). As an alternative, if we had chosen core patterns using state as the granularity level, the framework may not identify those patterns which are frequent in one city but infrequent in the others. To choose the minimum-support value, regarding the *large size* of data and potential *seasonality* in observations, we followed the approach proposed by Fournier-Viger [10]. Based on this approach, we used Equation 3 to find the minimum support, where a, b, and c are the positive constants which we empirically set to 0.004, 1.5, and 0.05, respectively. In this formula, x is the number of relation trees in a set, and the minimum relative support is 5%.

$$\text{min\_sup} = e^{-(ax+b)} + c \tag{3}$$

Using the above setting, we extracted 708 frequent tree patterns for the contiguous United States. In total, there were 90 unique frequent patterns, with the minimum number of nodes in a tree pattern being 2 and the maximum being 7. Figure 4 shows the top 12 frequent tree patterns[6]. Along with each pattern is shown the number of states which have occurrences of that pattern, the average support value, the peak time for instances of the pattern, and type of the road-network in which instances of a pattern were common. A road-network can be a road inside a city (cities), an interstate or freeway which connects different cities or states to each other, or a mixture of both. Each pattern shows how short-term entities are propagated in a region. For instance, pattern 1 shows a congestion which caused another congestion, and pattern 2 shows a propagation pattern of a chain of traffic congestion entities.

In total, 50 of 90 unique frequent patterns were initiated by a weather event, where 17 of these patterns were initiated by rain, 14 by snow, 11 by fog, and 8 by the other types of weather entities. These observations demonstrate the significant impact of weather on traffic. While this has been frequently discussed in prior research [2, 5, 6], in our work we reveal the propagation patterns which show HOW these weather entities impact traffic. For example, snow-initiated patterns usually happen on interstates and freeways, while rain-initiated patterns happen within roadways inside cities. Also, most complex congestion-related patterns happen within cities road-network, with the average support of patterns which happen in a city is lower than those which happen on interstates and freeways, or the entire road-network. The peak time for the majority of congestion-related patterns was the afternoon rush hour. For weather initiated patterns (except for the rain-related cases), the peak time was the morning rush hour. It was interesting to note that some weather events caused more traffic issues in the morning rather than the afternoon.

To further analyze the short-term patterns, we created a one-hot vector of size 90 for each state which represents the presence or absence of each unique short-term pattern. By applying K-means clustering [12] on these vectors, we categorized different states based on their short-term propagation patterns. To find the best number of clusters, we adapted Description Length (DL) for K-Means [9], which is represented by Equation 4. In this equation, $p(.)$ is the probability density function based on distance of each data point x from its cluster center c_x; P is the number of parameters of distribution function; K is the number of clusters; and X is the set of all data points. By assuming the distribution function for distance from cluster centers to be a Gaussian distribution, we have $P = 2$. By choosing K from set $[2, 3, \ldots, 10]$, we found the optimal number of clusters to be 4, which provides the minimum description length.

$$DL(K) = -\sum_{x \in X} log(p(||x - c_x||)) + \frac{1}{2} P\ log(|X|) + K\ log(|X|) \tag{4}$$

[5]For entity type *snow*, we empirically set $T\text{-}thresh = 40$ minutes, because we expect to see a longer impact of snow on traffic flow.

[6]Check https://bit.ly/2Ef8tu7 for the list of all short-term frequent patterns in our data.

Table 3: Clustering of 49 states into 4 clusters based on their short-term patterns, using K-Means.

Cluster	States
Cluster 1	AL, AR, CT, DC, DE, IA, IN, LA, MA, ME, MI, MN, MO, MS, NE, NH, OH, OK, RI, SD, TN, VT, WI
Cluster 2	AZ, CO, ID, KS, KY, MD, MT, NC, ND, NJ, NM, NV, OR, PA, SC, UT, VA, WV, WY
Cluster 3	FL, GA, IL, NY, TX, WA
Cluster 4	CA

Table 3 shows the result of clustering, in which we profile clusters as follows:

- **Cluster 1**: mostly contains states with fewer traffic incidents (as related to weather). These are either states with lower population (e.g., NE, SD, etc.); or states where the impact of weather is mitigated by effective road crews (e.g., OH, MN, etc.).
- **Cluster 2**: mostly contains states with considerably more traffic issues in comparison to the states in cluster 1. Distinguished patterns which only observed for this cluster are chain of accidents, and complex snow-initiated patterns.
- **Cluster 3**: contains states with at-least one major city with significant traffic issues. Distinguished patterns observed for this cluster are those which initiated by construction, rain, severe-cold, and storm.
- **Cluster 4**: contains only one state whose traffic patterns bore no similarity to any other state. Majority of distinguished patterns of this cluster are complex congestion-related, fog-initiated, and flow-incident related ones.

It is worth noting that the states which were clustered together were not necessarily located in the same geographical region, and might not have the same weather condition during the different seasons. However, their propagation patterns of traffic and impact of weather on traffic was found to be the same, which led to them being in the same cluster.

6.2 Long-term Pattern Discovery Results

6.2.1 Parameter Settings and Conventions. As described in Section 5.2, first we use the 99^{th} percentile of distribution of the duration of entities across the entire dataset, as the threshold to extract long entities – resulting in about 300 minutes. Using this threshold, we extracted 280,649 long entities. To merge the overlaps by Algorithm 1, we set $\rho = R$, where R is the spatial neighborhood distance to define sets $\mathcal{S}_R$, $\mathcal{S}\text{–}before_R$, and $\mathcal{S}\text{–}after_R$. In this way, we ensure that after the merge, there is no pair of long entities whose spatial neighborhood overlapped. Next we describe how to determine R, and then perform merging the overlaps.

Extracting R for long Traffic entities. To determine R, we use a random sample S_1 of two million traffic entities, and apply DBSCAN [12] to cluster entities in set S_1. We find the radius of each cluster as the maximum distance from the center, and obtain R as the average radius across all clusters. To define the two DBSCAN parameters – ϵ (maximum neighborhood distance) and *minPts* (the minimum required number of neighbors for not being an outlier), we use Algorithm 2 adapted from [23]. Using a random sample set of 0.5 million traffic entities in terms of S_2, we obtained $\epsilon = 4.09$ *miles* and $minPts = 463$. Applying DBSCAN on S_1 resulted in 191 clusters, with the average radius R of these clusters being 14.03 *miles*.

Note that we cannot quantitatively define R for long weather entities. Thus, we define a traffic entity t be within *R–neighborhood* of a long-term weather entity w, if t's zipcode can be mapped to the airport station which w is reported from, as the closest station. With $\rho = 14.03$, and after merging the overlaps, we ended up with 148,237 long entities. Table 4 provides the details on top-15 types of long entities, before and after the merge. Note that after the merge process, some of the types were combined to generate new type labels (e.g., Rain_Event). Next, setting $R = 14.03$, we created vicinity sets $\mathcal{S}_R$, $\mathcal{S}\text{–}before_R$, and $\mathcal{S}\text{–}after_R$ for each long-term entity.

Algorithm 2: Finding DBSCAN Parameters

1: Input: S_2, a large sample of traffic entities.
2: In S_2, obtain the closest neighbor distance for each entity, and let C_1 be the 99^{th} percentile of distribution of the closest neighbor distances.
3: For each entity, count the number of entities within distance C_1, and obtain distribution of count values over S_2.
4: Let C_2 be the 99^{th} percentile of distribution of the count values.
5: Output: C_1 as ϵ, and C_2 as $minPts$

Bucketing. Prior to employing statistical significance testing to identify long-term patterns, we need to determine the buckets of long-term entities. We use three different criteria to create disjoint buckets, namely, *Location*, *Duration*, and *Type*. Each "Location" bucket contains all the long entities which occurred in the same state. For the "Duration bucket", we first define several duration buckets (intervals), and then assign each long entity to a bucket. For "Type buckets", we create buckets of long entities, where each bucket contains all the entities of the same type.

Positive and Negative Impacts. The positive (negative) impact refers to the case where the value of $\mathcal{S}_R$ is larger (lower) than $\mathcal{S}\text{–}before_R$, $\mathcal{S}\text{–}after_R$, or their average. A significant positive impact can be determined by tests T_1, T_3, or T_5. A significant negative impact can be determined by tests T_2, T_4, or T_6.

Table 4: Top 15 long entity types and their frequency.

Before Merge		After Merge	
Type	Frequency	Type	Frequency
Construction	113,984	Rain	38,253
Rain	41,668	Snow	25,820
Event	32,144	Construction	22,373
Snow	27,723	Fog	19,553
Fog	20,847	Event	16,753
Congestion	17,314	Severe-Cold	11,671
Flow-Incident	13,099	Congestion	3,206
Severe-Cold	12,083	Flow-Incident	2,381
Storm	733	Construction_Event	1,332
Other	440	Construction_rain	758
Lane-Blocked	253	Storm	709
Accident	226	Congestion_Flow-Incident	675
Precipitation	72	Congestion_Event	516
Broken-Vehicle	49	Event_Rain	514
Hail	14	Congestion_Construction	359
total	280,649	total	144,873

6.2.2 Long-term Patterns. We present the identified long-term patterns in terms of three categories of such patterns; each category obtained based on a particular bucketing criteria.

Location-based Patterns. Using location as the bucketing criteria, we applied significance tests T_1 and T_2 to identify patterns of the form "*long-term entity in location L → more (or less) traffic incidents*". Figure 5 shows the results of these tests. Here we represent $1 - p\text{-}value$, and also show three confidence levels 90%, 95%, and 99% as three red lines (the results of other tests are not presented because any different trend of results was not observed). For a majority of states, we observed that a long-term weather/traffic entity had a significant impact on traffic flow. In the majority of the cases we found the result of test T_1 to be significant, which means a positive impact. Out of the 49 states, we found 30 to be significant with a confidence of 99%, 8 with a confidence of 95%, and 5 with a confidence of 90%. We also found that the existence of long-term traffic or weather entities

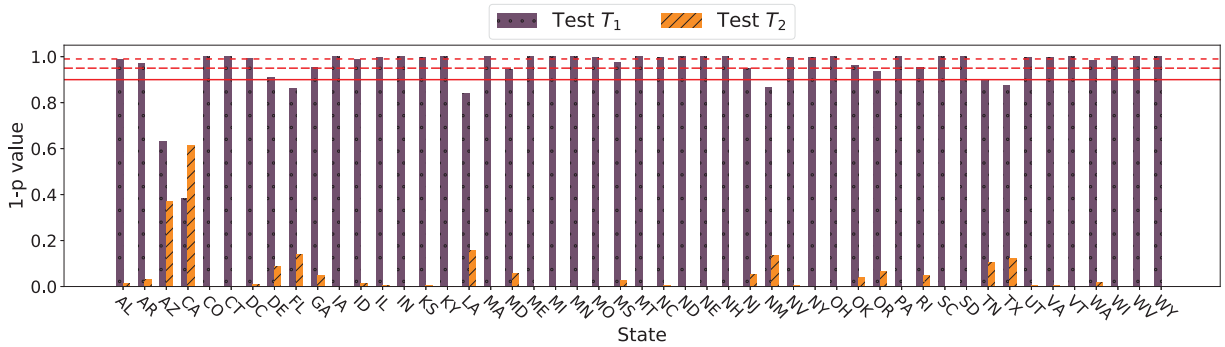
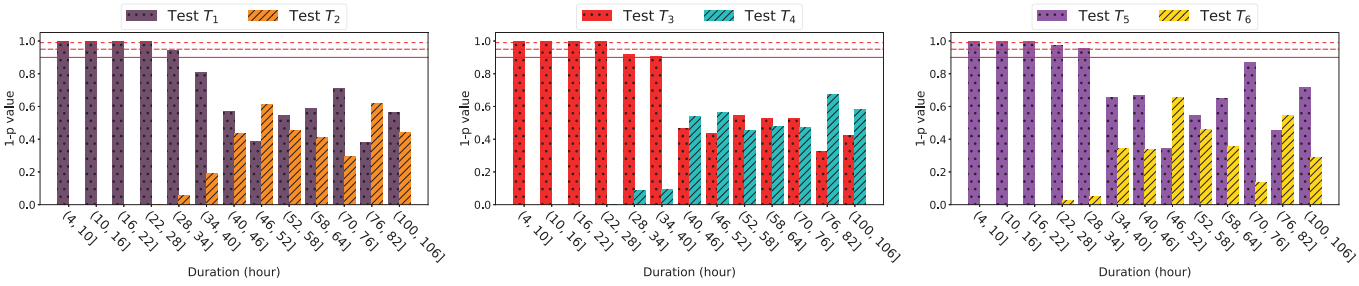
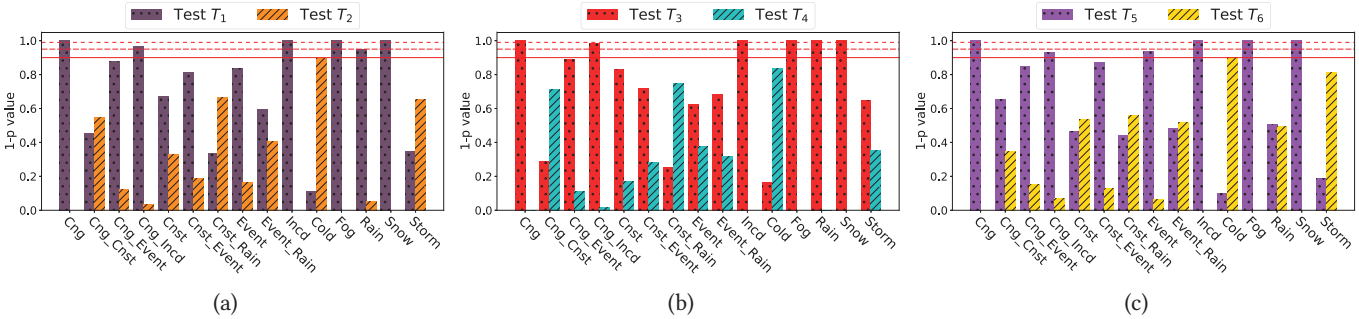


Figure 5: Statistical significance testing by test T_1 and T_2 for *Location* buckets. Red lines show three confidence levels 90%, 95%, and 99%.



(a) (b) (c)

Figure 6: Statistical significance testing for *Duration* buckets. Red lines show three confidence levels 90%, 95%, and 99%.



(a) (b) (c)

Figure 7: Statistical significance testing for *Type* buckets. Red lines show three confidence levels 90%, 95%, and 99%. Cng, Cnst, Incd, and Cold are short for Congestion, Construction, Flow-Incident, and Severe-Cold, respectively.

did not have much impact on traffic flow for AZ, CA, FL, LA, NM, and TX; although three of these states (i.e., CA, FL, and TX) are the top-3 states with the most observed traffic entities. This observation reveals that in a state with more traffic issues, the existence of a long-term incident does not have much impact on traffic flow. Incidentally, CA was the only state for which the p-value is found to be lower by Test T_2 (although insignificantly so). This could imply that CA has a unique condition where a long-term weather or traffic entity causes less traffic issues in comparison to the time when there is no such long-term entity.

Duration-based Patterns. Using duration of long entities as the bucketing criteria, we applied all the six significance tests to identify patterns of the form "*long-term entity with duration D → more (or less) traffic incidents*". Figure 6 shows the results of these tests. We conclude that the shorter the duration of a long-term entity is, the more significant its impact. Also, for long-term entities which lasted for more than 40 hours, we usually do not observe any significant impact. This observation might be due to adaptation of driving habits to the new conditions. Also, a comparison of the results of tests T_3 and T_4 with tests T_5 and T_6, provided evidences of more positive impacts based on the *after* interval, rather than the *before* interval, for long entities which lasted more than 28 hours. Given that a majority of such long entities were construction projects (about 75%), we posit two potential interpretations. First, after a long construction project, we tended to observe a smoother traffic flow, even in comparison to the time before the construction event. This observation might be due to the road conditions improving after the construction, but also could point to the fact that, after a long construction project, there might be a significant group of drivers who stuck with the alternative routes discovered when the construction was active.

Entity-type-based Patterns. Using type of the long entities as the bucketing criteria, we applied all the six significance tests to identify patterns of the form "*long-term entity of type T → more (or less) traffic incidents*". Figure 7 shows the results of these tests. Regarding the weather-based long entities, we observe the significant impact of all available types of weather entities, except for the "storm" event. However, we have an interesting diversity among impacts of different types of weather entities. Usually for "fog", "snow" and "rain", based on Tests T_1 and T_2, we see a positive impact on traffic, while for "severe-cold" we observe a negative impact. This

observation reveals that in extremely cold temperatures, we should expect to see smoother traffic flow probably because of fewer vehicles on the roads. Tests T_3 through T_6 also support such conclusion. Regarding the traffic-based long entities, we observed significant impacts by "congestion", "event", and "flow-incident". In case of a long-term "congestion", we have positive impact in comparison to before and after. For "flow-incident", we also observed a similar situation. However, for a long-term "event", we only observed positive impact in comparison to the time when the "event" is terminated (test T_5). It was interesting to note that a long-term construction had almost no significant impact on traffic flow. However, based on tests T_3 and T_4, we could expect to see more traffic issues during a long-term construction than before it or after it.

7 CONCLUSION AND FUTURE WORK

To overcome with the shortcomings posed by the existing general-purpose spatiotemporal pattern discovery frameworks, such as relying on a simplistic definition of spatiotemporal neighborhood, we present a new framework to extract *propagation* as well as *influential* patterns in geo-spatiotemporal data using improved and novel techniques. To extract propagation patterns, that indicate immediate impacts, we use a stricter definition of spatial collocation and co-occurrence relationships to create relation trees, and then perform tree pattern mining in a forest of relation trees. Influential patterns, that show lagging impacts, explore the impact of long-lived geospatial entities on their neighborhood, and we used statistical techniques to identify such patterns. Using a new and unique geo-spatiotemporal dataset of traffic and weather entities, which is collected, processed, and augmented for the contiguous United States over two years, we explored 90 prevalent propagation patterns, where 50 of them were initiated by weather (mostly observed in morning) and the rest by traffic entities (mostly observed in afternoon). Based on these patterns, we identified four categories of US states, which show similarity of driving behavior and transportation infrastructures between different states. We also studied the lagging impact of long-term traffic or weather entities, with respect to location, duration, and type of the entities. Interestingly, we identified a positive impact of long-term entities in a majority of the states, except a few ones such as CA, FL, and TX. In general, we found that long-term entities which lasted for at most 40 hours have the maximum impact on traffic flow. We found that long-term congestion, snow, rain, fog, severe-cold, and flow-incidents cause the most significant lagging impact on traffic flow. In terms of future research, we plan to separately study the lagging impact of different entity types for different states and top-cities.

ACKNOWLEDGMENT

This work is supported by a grant from the NSF (EAR-1520870) and one from the Ohio Supercomputer Center (PAS0536). Any findings and opinions are those of the authors. We also thank Mr. Abbas Shakiba for the preliminary discussions.

REFERENCES

[1] Anna K Andersson and Lee Chapman. 2011. The impact of climate change on winter road maintenance and traffic accidents in West Midlands, UK. *Accident Analysis & Prevention* 43, 1 (2011), 284–289.
[2] Tom Brijs, Dimitris Karlis, and Geert Wets. 2008. Studying the effect of weather conditions on daily crash counts using a discrete time-series model. *Accident Analysis & Prevention* 40, 3 (2008), 1180–1190.
[3] Mete Celik. 2015. Partial spatio-temporal co-occurrence pattern mining. *Knowledge and Information Systems* 44, 1 (2015), 27–49.
[4] Mete Celik, Shashi Shekhar, James P Rogers, and James A Shine. 2008. Mixed-Drove Spatio-Temporal Co-occurrence Pattern Mining. *network* 11 (2008), 15.
[5] Mario Cools, Elke Moons, and Geert Wets. 2010. Assessing the impact of weather on traffic intensity. *Weather, Climate, and Society* 2, 1 (2010), 60–68.
[6] Ye Ding, Yanhua Li, Ke Deng, Haoyu Tan, Mingxuan Yuan, and Lionel M Ni. 2017. Detecting and analyzing urban regions with high impact of weather change on transport. *IEEE Transactions on Big Data* 3, 2 (2017), 126–139.
[7] Daniel Eisenberg. 2004. The mixed effects of precipitation on traffic crashes. *Accident analysis & prevention* 36, 4 (2004), 637–647.
[8] SR Ely. 1990. RDS-ALERT: a DRIVE project to develop a proposed standard for the Traffic Message Channel feature of the radio data system RDS. In *Car and its Environment-What DRIVE and PROMETHEUS Have to Offer, IEE Colloquium on*. IET, 8–1.
[9] Erik Erlandson. 2016. http://erikerlandson.github.io/blog/2016/08/03/x-medoids-using-minimum-description-length-to-identify-the-k-in-k-medoids/. (2016). Accessed: 2019-01-31.
[10] Philippe Fournier-Viger. 2010. *Un modèle hybride pour le support à l'apprentissage dans les domaines procéduraux et mal définis*. Ph.D. Dissertation. Université du Québec à Montréal.
[11] Derrick Hambly, Jean Andrey, Brian Mills, and Chris Fletcher. 2013. Projected implications of climate change for road safety in Greater Vancouver, Canada. *Climatic Change* 116, 3-4 (2013), 613–629.
[12] Jiawei Han, Jian Pei, and Micheline Kamber. 2011. *Data mining: concepts and techniques*. Elsevier.
[13] Haversine. 2019. https://en.wikipedia.org/wiki/Haversine_formula. (2019). Accessed: 2019-01-31.
[14] Yan Huang, Shashi Shekhar, and Hui Xiong. 2004. Discovering colocation patterns from spatial data sets: a general approach. *IEEE Transactions on Knowledge and Data Engineering* 16, 12 (2004), 1472–1485.
[15] Ryo Inoue, Akihisa Miyashita, and Masatoshi Sugita. 2016. Mining spatio-temporal patterns of congested traffic in urban areas from traffic sensor data. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 731–736.
[16] David Jaroszweski and Tom McNamara. 2014. The influence of rainfall on road accidents in urban areas: A weather radar approach. *Travel behaviour and society* 1, 1 (2014), 15–21.
[17] Tanvi Jindal, Prasanna Giridhar, Lu-An Tang, Jun Li, and Jiawei Han. 2013. Spatiotemporal periodical pattern mining in traffic data. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*. ACM, 11.
[18] Sachin Kumar and Durga Toshniwal. 2016. A data mining approach to characterize road accident locations. *Journal of Modern Transportation* 24, 1 (2016), 62–72.
[19] 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*. ACM, 1010–1018.
[20] Pradeep Mohan, Shashi Shekhar, James A Shine, and James P Rogers. 2010. Cascading spatio-temporal pattern discovery: A summary of results. In *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM, 327–338.
[21] Pradeep Mohan, Shashi Shekhar, James A Shine, and James P Rogers. 2012. Cascading spatio-temporal pattern discovery. *IEEE Transactions on Knowledge and Data Engineering* 24, 11 (2012), 1977–1992.
[22] Nominatim. 2019. https://wiki.openstreetmap.org/wiki/Nominatim. (2019). Accessed: 2019-01-31.
[23] DBSCAN parameter tunning. 2019. https://github.com/alitouka/spark_dbscan/wiki/Choosing-parameters-of-DBSCAN-algorithm/. (2019). Accessed: 2019.
[24] Feng Qian, Qinming He, and Jiangfeng He. 2009. Mining spread patterns of spatio-temporal co-occurrences over zones. In *International Conference on Computational Science and Its Applications*. Springer, 677–692.
[25] Shashi Shekhar, Zhe Jiang, Reem Y Ali, Emre Eftelioglu, Xun Tang, Venkata Gunturi, and Xun Zhou. 2015. Spatiotemporal data mining: a computational perspective. *ISPRS International Journal of Geo-Information* 4, 4 (2015), 2306–2338.
[26] Shirish Tatikonda and Srinivasan Parthasarathy. 2009. Mining Tree-Structured Data on Multicore Systems. *PVLDB* 2, 1 (2009), 694–705.
[27] Shirish Tatikonda, Srinivasan Parthasarathy, and Tahsin Kurc. 2006. TRIPS and TIDES: new algorithms for tree mining. In *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 455–464.
[28] Athanasios Theofilatos. 2017. Incorporating real-time traffic and weather data to explore road accident likelihood and severity in urban arterials. *Journal of safety research* 61 (2017), 9–21.
[29] Athanasios Theofilatos and George Yannis. 2014. A review of the effect of traffic and weather characteristics on road safety. *Accident Analysis & Prevention* 72 (2014), 244–256.
[30] Faizan Wajid and Hanan Samet. 2016. Crimestand: Spatial tracking of criminal activity. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 81.
[31] Ling Wang, Qi Shi, and Mohamed Abdel-Aty. 2015. Predicting crashes on expressway ramps with real-time traffic and weather data. *Transportation Research Record: Journal of the Transportation Research Board* 2514 (2015), 32–38.
[32] MapQuest website. 2014-2018. https://www.mapquest.com/. (2014-2018). Accessed: 2019-01-31.
[33] Hui Yang, Srinivasan Parthasarathy, and Sameep Mehta. 2005. A generalized framework for mining spatio-temporal patterns in scientific data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 716–721.
[34] Wenhao Yu. 2016. Spatial co-location pattern mining for location-based services in road networks. *Expert Systems with Applications* 46 (2016), 324–335.
[35] Mohammed J Zaki. 2005. Efficiently mining frequent embedded unordered trees. *Fundamenta Informaticae* 66, 1-2 (2005), 33–52.