Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights

Sobhan Moosavi
The Ohio State University
Columbus, Ohio
moosavi.3@osu.edu

Mohammad Hossein Samavatian
The Ohio State University
Columbus, Ohio
samavatian.1@osu.edu

Srinivasan Parthasarathy
The Ohio State University
Columbus, Ohio
srini@cse.ohio-state.edu

Radu Teodorescu
The Ohio State University
Columbus, Ohio
teodores@cse.ohio-state.edu

Rajiv Ramnath
The Ohio State University
Columbus, Ohio
ramnath@cse.ohio-state.edu

ABSTRACT

Reducing traffic accidents is an important public safety challenge, therefore, accident analysis and prediction has been a topic of much research over the past few decades. Using small-scale datasets with limited coverage, being dependent on extensive set of data, and being not applicable for real-time purposes are the important shortcomings of the existing studies. To address these challenges, we propose a new solution for real-time traffic accident prediction using easy-to-obtain, but sparse data. Our solution relies on a deep-neural-network model (which we have named *DAP*, for Deep Accident Prediction); which utilizes a variety of data attributes such as *traffic events*, *weather data*, *points-of-interest*, and *time*. DAP incorporates multiple components including a recurrent (for time-sensitive data), a fully connected (for time-insensitive data), and a trainable embedding component (to capture spatial heterogeneity). To fill the data gap, we have - through a comprehensive process of data collection, integration, and augmentation - created a large-scale publicly available database of accident information named *US-Accidents*. By employing the US-Accidents dataset and through an extensive set of experiments across several large cities, we have evaluated our proposal against several baselines. Our analysis and results show significant improvements to predict rare accident events. Further, we have shown the impact of traffic information, time, and points-of-interest data for real-time accident prediction.

CCS CONCEPTS

• **Theory of computation → Data integration**; • **Computing methodologies → Supervised learning by classification**; • **Applied computing →** *Transportation*.

KEYWORDS

Accident Prediction, US-Accidents, Heterogeneous Data

SIGSPATIAL '19, November 5–8, 2019, Chicago, IL, USA

ACM ISBN 978-1-4503-6909-1/19/11…$15.00
https://doi.org/10.1145/3347146.3359078

ACM Reference Format:
Sobhan Moosavi, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu Teodorescu, and Rajiv Ramnath. 2019. Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights. In *27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '19), November 5–8, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3347146.3359078

1 INTRODUCTION

Reducing traffic accidents is an important public safety challenge around the world. A global status report on traffic safety [28] notes that there were 1.25 million traffic deaths in 2013 alone, with deaths increasing in 68 countries when compared to 2010. Accident prediction is important for optimizing public transportation, enabling safer routes, and cost-effectively improving the transportation infrastructure, all in order to make the roads safer. Given its significance, accident analysis and prediction has been a topic of much research in the past few decades. Analyzing the impact of environmental stimuli (e.g., road-network properties, weather, and traffic) on traffic accident occurrence patterns [10, 15, 30], predicting frequency of accidents within a geographical region [3, 6, 23, 29, 36], and predicting risk of accidents [8, 18, 35, 37] are the major related research categories.

Employing small-scaled datasets with limited coverage (e.g. a small number of road-segments, or just one city) [3, 5, 6, 18, 35]; being dependent on a wide range of data attributes which may not be available for all regions (e.g., satellite imagery, traffic volume, and properties of road-network) [23, 36, 37]; being not applicable for real-time applications regarding the modeling constraints and prerequisites (e.g., prediction for longer time intervals such as one day or one week, or requiring extensive set of data) [3, 23, 29, 36]; and employing over simplified methods for traffic accident prediction [3, 13, 18] are the main shortcomings of the existing studies.

To address these challenges and provide a reasonable solution for real-time traffic accident prediction, we propose *DAP*, a deep-neural-network-based accident prediction model. DAP uses a variety of data including *traffic events* (e.g., congestion, construction, and road hazards), *weather* (e.g., temperature, visibility, and wind speed), *points-of-interest* (e.g., traffic signal, stop sign, and junction), and *time* (e.g., day of week, hour of day, and period of day) to provide real-time prediction for a geographical region of reasonable size (i.e., a square of size $5km \times 5km$ on map) and during a fine-grained time period (i.e., a 15 minutes interval). To our knowledge, this is the

first research work that has employed traffic events and points-of-interest data for accident prediction. DAP exerts multiple important components to utilize different categories of attributes. To utilize time-sensitive data (e.g., traffic, weather, and time data), DAP employs a recurrent component with Long Short Term Memory (LSTM) cells. To utilize time-insensitive data (e.g., points-of-interest), DAP employs feed-forward neural network layers. Further, to better capture spatial heterogeneity, which has been proven to be effective for accident prediction [37], DAP employs trainable latent representation for each geographical region to encode essential spatiotemporal information.

In order to mitigate the impact of data size on analysis and prediction, we present a new dataset, we name it *US-Accidents*, which includes about 2.25 million traffic accidents took place within the contiguous United States[1], between February 2016 and March 2019. US-Accidents offers a wide range of data attributes to describe each accident including *location data*, *time data*, *natural language description of event*, *weather data*, *period-of-day information*[2], and *relevant points-of-interest data*. Importantly, we also present our *process* for creating the above dataset from streaming traffic reports and heterogeneous contextual data (weather, points-of-interests, etc.), so that the community can validate it, and with the belief that this process can itself serve as a model for dataset creation. We performed a variety of data analysis and profiling based on US-Accidents dataset to derive a wide-range of insights. Our analyses demonstrated that about 40% of accidents took place on or near high-speed roadways (highways, interstates, etc.) and about 32% on or near local roads (streets, avenues, etc.). We also derived various insights with respect to the correlation of accidents with time, points-of-interest, and weather conditions.

Using US-Accidents, and through extensive experiments across several large cities, we compared our proposal against several neural-network-based and traditional machine learning models (such as logistic regression and gradient boosting classifier). Our analysis and results show the superiority of our model in terms of improvement of $f1\text{-}score$ for the case of positive examples (i.e., cases which labeled as accident), by about 16% in comparison to the best traditional model, and about 7% in comparison to the best neural-network-based model. When considering both positive and negative cases (negative cases are labeled as non-accident, which are the majority), our proposal achieves comparable results when compared to the best baselines. Nevertheless, we note that positive cases are far more important, regarding their rare nature, and importance to be properly predicted. Further, we conducted thorough analyses to assess the ability of different categories of attributes for real-time traffic accident prediction using multiple testing scenarios. Our findings indicate the importance of time, points-of-interest, and traffic data for this task.

The main contributions of this paper are therefore as follows.

- A new methodology for heterogeneous data collection, cleansing, and augmentation to prepare a unique, large-scale dataset of traffic accidents. This dataset has been collected for the contiguous United States over three years, and contains 2.25 million traffic accidents. The dataset is publicly available for the research community at https://smoosavi.org/datasets/us_accidents.
- A variety of insights gleaned through analyses of accident hotspot locations, time, weather and points-of-interest correlations with the accident data. These insights may directly be utilized for applications such as urban planning, exploring flaws in transportation infrastructure design, traffic control and prediction, and personalized insurance.
- A new deep-neural-network-based solution for traffic accident prediction using heterogeneous sparse data. To the best of our knowledge, this is the first work which uses information from *traffic flow*, fused with other available sources of contextual data such as "weather" and "points-of-interest", to perform accident prediction. Furthermore, our methodology predicts future accidents at the fine-grained time interval of 15 minutes.

For the rest of this paper, we first provide preliminaries in Section 3. The overview of related work is discussed in Section 2. Section 4 describes the dataset construction process and the resulting dataset. The accident prediction framework is presented in Section 5, followed by experiments and results in Section 6. Finally, Section 7 concludes the paper.

2 RELATED WORK

Accident analysis and prediction has been the topic of many research during the past few decades, where we study three categories of these work as follows.

Analysis of Environmental Stimuli on Accidents. This category of work investigates the impact of environmental stimuli (e.g., weather, traffic flow, and properties of road-network) on possibility or severity of traffic accidents. Studying the impact of weather factors (e.g., precipitation) on road accidents [10, 15, 30, 31]; applying data mining techniques to extract association rules to perform causality analysis [1, 17]; and statistical analysis of unobserved heterogeneity to explore the impact of unavailable variables (e.g., missing data) on severity of traffic accidents [19] are some examples of this category. These studies usually provide significant insights, however, may not be directly utilized for real-time prediction and planning.

Accident Frequency Prediction. Prediction of the expected number of traffic accidents for a specific road-segment or geographical region is the target of this group of studies [6]. Early work in this area by Chang et al. [5] used information such as road geometry, annual average daily traffic (AADT), and weather data to predict the frequency of accidents for a highway using a neural network model. Caliendo et al. [3] used a set of road-related attributes such as length, curvature, AADT, sight distance, and presence of junction to predict frequency of accidents. The usage of satellite imagery to predict the frequency of accidents by a convolutional neural network model using large scale accident and imagery data was proposed by Najjar et al. [23]. Further, Ren et al. [29] recently used a Long Short Term Memory (LSTM) model to predict the frequency of accidents, given the history of past 100 hours, for grid cells of size $1km \times 1km$. Similarly, Chen et al. [7] proposed to use a stack denoising convolutional autoencoder model to predict frequency of accidents for grid cells using traffic flow (collected using plate recognition systems), past traffic accidents, and time data. Yuan et al. [36] proposed hetero-ConvLSTM to predict frequency of traffic accidents using several sources of environmental data such as traffic volume, road condition, rainfall, temperature, and satellite images. They evaluated their model using a large-scale data of traffic accidents from state of Iowa, performed predictions for grid cells of size $5km \times 5km$, and showed the importance of capturing spatial heterogeneity and temporal trends to better predict traffic accidents

[1]The contiguous United States excludes Alaska and Hawaii, and considers District of Columbia (DC) as a separate state.
[2]Period-of-day is associated with daylight, thus it is represented as "day" or "night".

[36]. Studies in this category usually make use of many pieces of information that may not be available in real-time applications.

Accident Risk Prediction. This category of work is very much similar to the previous one, unless prediction here is defined as a binary classification task which better fits real-time applications [35, 37]. Using data for a single segment of I-64 in Virginia (US), Lin et al. [18] leveraged a decision tree model to separate pre-crash records from normal ones, using information such as weather, visibility, traffic volume, speed, and occupancy information. However, their limited size of data might weaken their solution or findings. In another study Chen et al. [8] used human mobility data in terms of 1.6 million GPS records and a set of 300,000 accident records in Tokyo (Japan) to predict the possibility of accident occurrence on grid cells of size $500m \times 500m$ in an hourly basis. They leveraged a stack denoising autoencoder model to extract latent features from human mobility, and then used a logistic regression model to predict accidents. Finally, Yuan et al. [37] used a heterogeneous set of urban data such as road characteristics (AADT, speed limit, etc.), radar-based rainfall data, temperature data, and demographic data to predict probability of accident for each road-segment in state of Iowa. They leveraged eigen-analysis to capture and represent spatial heterogeneity. Their analyses and results suggest the importance of time, human factors, weather data, and road-network characteristics for this task.

Our proposal belongs to the last category as we seek to perform accident risk prediction. Further, our solution is more suitable for real-time applications as we provide prediction for much shorter time interval (i.e., 15 minutes) in comparison to literature. Besides, our usage of real-time traffic events and points-of-interest, to the best of our knowledge, is not discussed before. Lastly, the type of input data which we use for prediction is rather easy to collect and available to public, in contrast to those work which used extensive set of data for modeling and prediction.

3 PRELIMINARIES AND PROBLEM STATEMENT

Definition 3.1 (Traffic Event)**.** We define a traffic event e by $e = \langle lat, lng, time, type, desc \rangle$, where lat and lng represent the GPS coordinates, $type$ is a categorical classification of the event, and $desc$ provides a natural language description of the event. A traffic event is one of the following types: *accident*, *broken-vehicle*, *congestion*, *construction*, *event*, *lane-blocked*, and *flow-incident*. Table 1 describes these events.

Table 1: Definition of Traffic Events.

Type	Description
Accident	A collision event which may involve one or more vehicles.
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 or speed-limit.
Construction	Refers to maintenance project on a road.
Event	Situations such as *sports event*, *demonstrations*, or *concerts*, that could potentially impact traffic flow.
Lane-blocked	Refers to the cases when we have blocked lane(s) due to traffic or weather condition.
Flow-incident	Refers to all other types of traffic events. Examples are *broken traffic light* and *animal in the road.*

Definition 3.2 (Weather Observation Record)**.** A weather observation w is defined by $w = \langle lat, lng, time, temperature, humidity, pressure, visibility, wind\text{-}speed, precip, rain, snow, fog, hail \rangle$. Here lat and lng represent the GPS coordinates of the weather station which reported w; $precip$ is the precipitation amount (if any); and rain, snow, fog, and hail are binary indicators of these events.

Definition 3.3 (Point-of-Interest)**.** A point-of-interest p is defined by $p = \langle lat, lng, type \rangle$. Here, lat and lng show the GPS latitude and longitude coordinates, and available types for p are described in Table 2. Note that several of definitions in this table are adopted from https://wiki.openstreetmap.org.

Table 2: Definition of Point-Of-Interest (POI) annotation tags based on Open Street Map (OSM).

Type	Description
Amenity	Refers to particular places such as restaurant, library, college, bar, etc.
Bump	Refers to speed bump or hump to reduce the speed.
Crossing	Refers to any crossing across roads for pedestrians, cyclists, etc.
Give-way	A sign on road which shows priority of passing.
Junction	Refers to any highway ramp, exit, or entrance.
No-exit	Indicates there is no possibility to travel further by any transport mode along a formal path or route.
Railway	Indicates the presence of railways.
Roundabout	Refers to a circular road junction.
Station	Refers to public transportation station (bus, metro, etc.).
Stop	Refers to stop sign.
Traffic Calming	Refers to any means for slowing down traffic speed.
Traffic Signal	Refers to traffic signal on intersections.
Turning Loop	Indicates a widened area of a highway with a non-traversable island for turning around.

Definition 3.4 (Geographical Region)**.** We define a geographical region r as a square of size $l \times l$ over the map of a city. The choice of l is related to application domain, and in this work we set $l = 5km$.

Given the preliminaries, we formulate the problem as follows:

Given:
- A spatial grid $\mathrm{R} = \{r_1, r_2, \ldots, r_n\}$, where each $r \in \mathrm{R}$ is a geographical region of size $5km \times 5km$.
- A set of fixed-length time intervals $\mathrm{T} = \{t_1, t_2, \ldots, t_m\}$, where we set $|t| = 15$ *minutes*, for $t \in \mathrm{T}$.
- A database of traffic events $E_r = \{e_1, e_2, \ldots\}$ for each geographical region $r \in \mathrm{R}$.
- A database of weather observation records $W_r = \{w_1, w_2, \ldots\}$ for each geographical region $r \in \mathrm{R}$.
- A database of points of interest $P_r = \{p_1, p_2, \ldots\}$ for each geographical region $r \in \mathrm{R}$.

Create:
- A representation F_{rt} for a region $r \in \mathrm{R}$ during a time interval $t \in \mathrm{T}$, using E_r, W_r, and P_r.
- A binary label L_{rt} for F_{rt}, where 1 indicates at least one traffic accident happened during t in region r, and 0 otherwise.

Find:
- A model $\mathcal{M}$ to predict L_{rt} using $\langle F_{rt_{i-8}}, F_{rt_{i-7}}, \ldots, F_{rt_{i-1}} \rangle$, which means predicting the label of current time interval using observations from the last 8 time intervals to

Objective:
- Minimize the prediction error.

4 ACCIDENT DATASET

This section describes the process of constructing a country-wide traffic accident dataset, which we named *US-Accidents*. An overview of this process is shown in Figure 1. US-Accident contains 2.25 million cases of traffic accidents that took place within the United States from February 2016 to March 2019. The following sub-sections provide a detailed description of each step of the data preparation process. The dataset is publicly available at https://smoosavi.org/datasets/us_accidents.

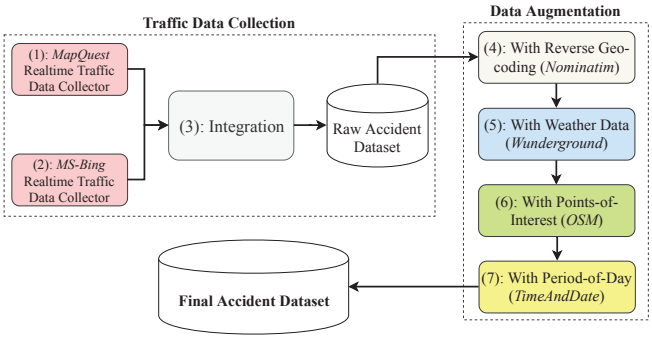


Figure 1: Process of Creating Traffic Accident Dataset

4.1 Traffic Data Collection

4.1.1 Realtime Traffic Data Collection. We collected streaming traffic data using two real-time data providers, namely "MapQuest Traffic" [20] and "Microsoft Bing Map Traffic" [2], whose APIs broadcast traffic events (accident, congestion, etc.) captured by a variety of entities - the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road-networks. We pulled data every 90 seconds from 6am to 11pm, and every 150 seconds from 11pm to 6am. In total, we collected 2.27 million cases of traffic accidents between February 2016 and March 2019; 1.73 million cases were pulled from MapQuest, and 0.54 million cases from Bing.

4.1.2 Integration. Integration of the data consisted of removing cases duplicated across the two sources and building a unified dataset. We considered two events as duplicates if their Haversine distance and their recorded times of occurrence were both below a heuristic threshold (set empirically at 250 meters and 10 minutes, respectively). We believe these settings to be conservative, but we settled on them in order to ensure a very low possibility of duplicates. Using these settings, we found about 24,600 duplicated accident records, or about 1% of all data. The final dataset after removing the duplicated cases comprised 2.25 million accidents.

4.2 Data Augmentation

4.2.1 Augmenting with Reverse Geo-Coding. Raw traffic accident records contained only GPS data. We employed the *Nominatim* tool [24] to perform reverse geocoding to translate GPS coordinates to addresses, each consisting of a *street number*, *street name*, *relative side (left/right)*, *city*, *county*, *state*, *country*, and *zip-code*. This process is same as *point-wise map-matching*.

4.2.2 Augmenting with Weather Data. Weather information provides important context for traffic accidents. Thus, we employed *Weather Underground* API [34] to obtain weather information for each accident. Raw weather data was collected from 1,977 weather stations located in airports all around the United States. 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 collected several data records per day, each of which was reported upon any significant change in any of the measured weather attributes.

Each traffic event e was augmented with weather data as follows. First the closest weather station s was identified. Then, of the weather observation records which were reported from s, we looked for the weather observation record w whose reported time was closest to the start time of e, and augmented it with weather data. In our integrated accident dataset, the average difference in report time for an accident record and its paired weather observation record was about 15 minutes.

4.2.3 Augmenting with Points-Of-Interest. Points-of-interest (POI) are locations annotated on a map as *amenities*, *traffic signals*, *crossings*, etc. These annotations are associated with *nodes* on a road-network. A node can be associated with a variety of POI types, however, in this work we only use 13 types as described in Table 2. We obtained these annotations from Open Street Map (OSM) [25] for the United States, using its most recently released dataset (extracted on April 2019). The applicable POI annotations for a traffic accident a are those which are located within a distance threshold τ from a. We determine this threshold by evaluating different values to find the value that is best able to associate a POI with an accident. Essentially, the objective is to find the best distance for which a POI annotation can be identified as *relevant* to an accident record. Therefore, we need a mechanism to measure the relevancy. To begin with, we note that the natural language descriptions of traffic accidents follow a set of regular expression patterns, and that a few of these patterns may be used to identify and use as an annotation for the location type (e.g., intersection or junction) of the accident.

Regular Expression Patterns. Given the description of traffic accidents, we were able to identify 27 regular expression patterns; 16 of them were extracted based on MapQuest data, and 11 from Bing data. Among the MapQuest patterns, the following expression corresponds to *junctions* (see Table 2): "... **on** ... **at exit** ...", and the following pattern mostly[4] determines an *intersection*: "... **on** ... **at** ...". An intersection is associated with *crossing*, *stop*, or *traffic signal* (see Table 2). Among Bing regular expression patterns, two of them identify junctions: "**at** ... **exit** ..." and "**ramp to** ...". Table 3 shows several examples of accidents, where the regular expression pattern (in bold face) identifies the correct POI type[5].

The essential idea is to find a threshold value that maximizes the correlation between annotations from POI and annotations derived using regular expression patterns. Thus, for a set of accident records, we annotate their location based on both methods, regular expression patterns as well as OSM-based POI annotations (using a specific distance threshold). Then, we measure the correlation between the annotations derived from these methods to find which threshold value provides the highest correlation (i.e., the best choice). Note that we employ the regular expression patterns as *pseudo* ground truth labels, to evaluate OSM-based POI annotations using different

[3] Possible values are *clear*, *snow*, *rain*, *fog*, *hail*, and *thunderstorm*.
[4] Using 200 randomly sampled accidents cases which were manually checked on a map, about 78% of matches using this pattern were actually occurred on intersections.
[5] These cases were manually checked on a map to ensure the correctness of the annotation.

Table 3: Examples of traffic accidents with their *annotation type* assigned using their natural language description by regular expression patterns.

Source	Description	Type
MapQuest	Serious accident **on** 4th Ave **at** McCullaugh Rd.	Intersection
MapQuest	Accident **on** NE-370 Gruenther Rd **at** 216th St.	Intersection
MapQuest	Accident **on** I-80 **at Exit** 4A Treasure Is.	Junction
MapQuest	Accident **on** I-87 I-287 Southbound **at Exit** 9 I-287.	Junction
Bing	**At** Porter Ave/**Exit** 9 - Accident. Left lane blocked.	Junction
Bing	**At** IL-43/Harlem Ave/**Exit** 21B - Accident.	Junction
Bing	**Ramp to** I-15/Ontario Fwy/Cherry Ave - Accident.	Junction
Bing	**Ramp to** Q St - Accident. Right lane blocked.	Junction

Algorithm 1: Find Annotation Correlation

1: Input: a dataset of traffic accidents $\mathcal{A}$, a database of points-of-interest $\mathcal{P}$, and a distance threshold τ.
2: Extract and create a set of regular expression patterns RE to identify a specific POI v.
3: Create set S_1: for each traffic accident $a \in \mathcal{A}$, we add it to S_1 if its natural language description $a.desc$ can be matched with at least one regular expression in set RE.
4: Create set S_2: for each traffic accident $a \in \mathcal{A}$, we add it to S_2 if there is at least one POI $p \in \mathbf{P}$ of type v, where $haversine\_distance\,(a, p) \leq \tau$.
5: Output: Return $\mathit{Jaccard}\,(S_1, S_2)$.

threshold values. We propose Algorithm 1 to find the best distance threshold. We use a sample of 100,000 accidents as set $\mathcal{A}$ (step 1). For step 2, we consider either "intersection" or "junction", and use the set of relevant regular expressions (see Table 3) in terms of RE. Next we create set S_1 by annotating each traffic accident $a \in \mathcal{A}$ using the regular expression patterns in RE (step 3). Then we annotate each traffic accident $a \in \mathcal{A}$ based on points-of-interests in $\mathcal{P}$, using the distance threshold τ to create S_2 (step 4). Finally, we calculate the Jaccard similarity score using Equation 1 (step 5):

$$\mathit{Jaccard}\,(S_1, S_2) = \frac{\left|S_1 \cap S_2\right|}{\left|S_1 \cup S_2\right|} \tag{1}$$

We examined the following candidate set to find the optimal threshold value (all values in meters): {5, 10, 15, 20, 25, 30, 40, 50, 75, 100, 125, 150, 200, 250, 300, 400, 500}. We separately studied samples from Bing and MapQuest, and employed corresponding regular expression patterns for "intersection" and "jucntion". Figure 2 shows the results for each data source and each annotation type. From Figure 2a, we see that the maximum correlation for intersections is obtained for a threshold value of 30 meters. Figures 2b and 2c show that 100 meters is an appropriate distance threshold for annotating a junction.

Thresholds for the other available annotations in Table 2 are derived from the thresholds for junction and intersection as described below:

- **Junction-based threshold.** Given the definition of a junction (i.e., a highway ramp, exit, or entrance), we used the same threshold (100 meters) for the following types: amenity and no-exit.
- **Intersection-based threshold.** Given the definition of an intersection, we used the same threshold (30 meters) for the following annotation types: bump, crossing, give-way, railway, roundabout, station, stop, traffic calming, traffic signal, and turning loop.

Using these thresholds, we augmented each accident record with points-of-interest. In summary, 27.5% of accident records were augmented with at least one of the available POI types in Table 2. Further discussion on annotation results are presented in Section 4.3.

4.2.4 Augmenting with Period-of-Day. Given the start time of an accident record, we used "TimeAndDate" API [32] to label it as *day* or *night*. We assign this label based on four different daylight systems, namely *Sunrise/Sunset*, *Civil Twilight*, *Nautical Twilight*, and *Astronomical Twilight*. Note that these systems are defined based on the position of the sun with respect to the horizon, and each provide a different definition for period-of-day[6].

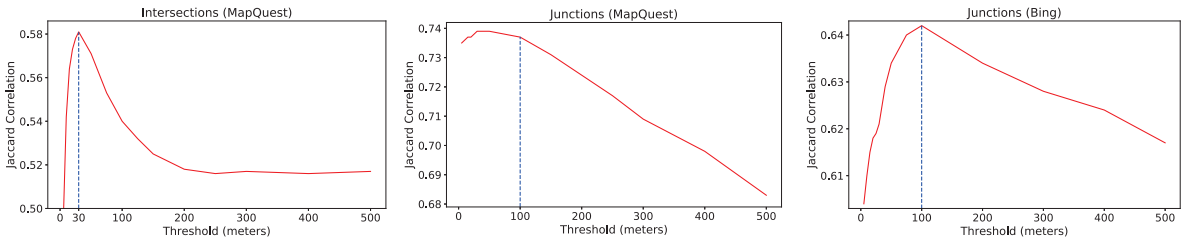
4.3 US-Accidents Dataset

Using the process described above, we created a countrywide dataset of traffic accidents, which we name *US-Accidents*. US-Accident contains about 2.25 million cases of traffic accidents that took place within the contiguous United States from February 2016 to March 2019. Table 4 shows the important details of US-Accidents. Also, Figure 3 provides more details on characteristics of the dataset. Figure 3-(a) shows that significantly more accidents were observed during the weekdays than weekends. Based on parts (b) and (c) of Figure 3, it can be observed that the hourly distribution during weekdays has two peaks (8am and 5pm), while the weekend distribution shows a single peak (1pm). Figure 3-(d) demonstrates that most of the accidents took place near junctions or intersections (crossing, traffic signal, and stop). MapQuest tends to report more accidents near intersections, while Bing reported more cases near junctions. This shows the complementary behavior of these APIs, and hence the comprehensiveness of our dataset. Figure 3-(e) describes distribution of road types, extracted from the map-matching results (i.e., street names). We used street names to identify type of the road. Here we note that about 32% of accidents happened on or near local roads (e.g., streets, avenues, and boulevards), and about 40% took place on or near high-speed roads (e.g., highways, interstates, and state roads). We also note that Bing reported more cases on high-speed roads. Finally, the period-of-day data shows that about 73% of accidents happened after sunrise (or during the day).

Table 4: US-Accidents: details as of March 2019.

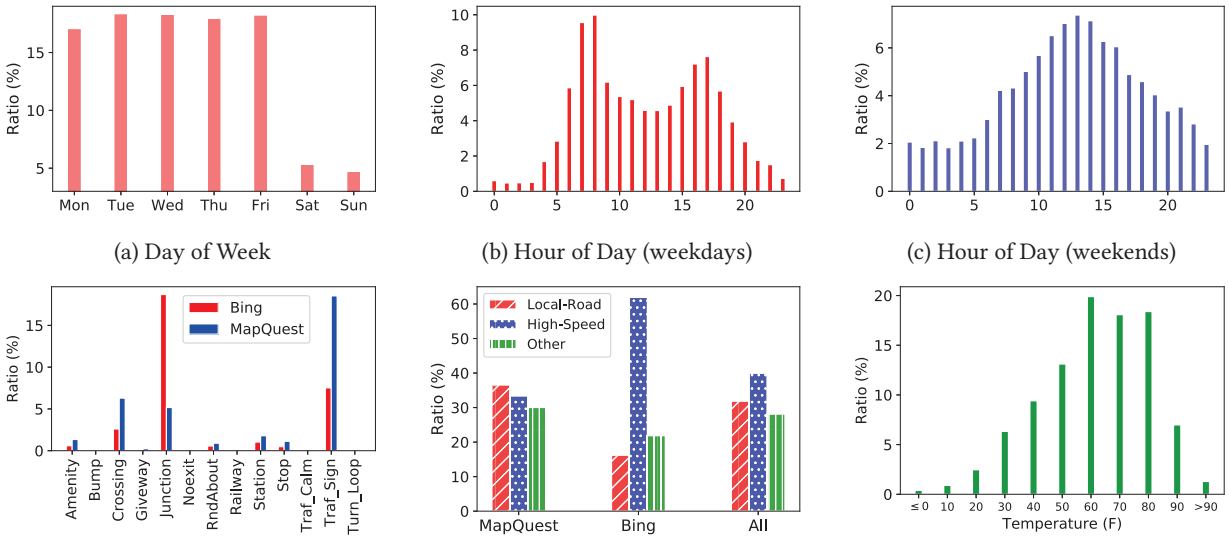
Total Attributes	45
Traffic Attributes (10)	id, source, TMC [33], severity, start_time, end_time, start_point, end_point, distance, and description
Address Attributes (8)	number, street, side (left/right), city, county, state, zip-code, country
Weather Attributes (10)	time, temperature, wind_chill, humidity, pressure, visibility, wind_direction, wind_speed, precipitation, and condition (e.g., rain, snow, etc.)
POI Attributes (13)	All cases in Table 2
Period-of-Day (4)	Sunrise/Sunset, Civil Twilight, Nautical Twilight, and Astronomical Twilight
Total Accidents	2,243,939
# MapQuest Accidents	1,702,565 (75.9%)
# Bing Accidents	516,762 (23%)
# Reported by Both	24,612 (1.1%)
Top States	California (485K), Texas (238K), Florida (177K), North Carolina (109K), New York (106K)

[6] See https://en.wikipedia.org/wiki/Twilight for more details.



(a) Using MapQuest for Intersection (b) Using MapQuest for Junction (c) Using Bing for Junction

Figure 2: Correlation study between regular-expression and OSM-based extracted annotations to find the best distance threshold values.



(a) Day of Week (b) Hour of Day (weekdays) (c) Hour of Day (weekends)

(d) points-of-interest Distribution (e) Road-type Distribution (f) Temperature Distribution

Figure 3: Characteristics of US-Accidents dataset, in terms of time analysis (a)–(c), points-of-interest-based augmentation distribution analysis (d), map-matching-based road type coverage analysis (e), and temperature analysis (f).

5 ACCIDENT PREDICTION MODEL

In this section we describe our traffic accident prediction framework. We start with description of feature vector representation, and then present our proposal for real-time traffic accident prediction.

5.1 Feature Vector Representation

Regarding the problem description in Section 3, we create a feature vector representation for each geographical region r of size $5km \times 5km$ during a time interval $t = 15\,minutes$. Such representation includes the following feature categories:

- **Traffic**: a vector of size 7 representing frequency of available traffic events (i.e., accident, broken-vehicle, congestion, construction, event, lane-blocked, and flow-incident) during the current 15 minutes interval. We obtain traffic events from [22].
- **Time**: includes *weekday* (a binary value to show weekday or weekend), *hour-of-day* (a one-hot vector of size 5 to show belonging to a specific time interval as defined in [21])[7], and *daylight* (an attribute to show period-of-day: day or night). We obtain daylight data from [32].
- **Weather**: a vector representing 10 weather attributes including temperature, pressure, humidity, visibility, wind-speed, precipitation amount; and four indicator flags for special events rain, snow, fog, and hail. We obtain weather data from [34].
- **POI**: a vector of size 13 to represent frequency of POIs within r, for amenity, speed bump, crossing, give-way sign, junction, no-exit sign, railway, roundabout, station, stop sign, traffic calming, traffic signal, and turning loop. We obtain POI data from [25].
- **Desc2Vec**: given a historical set of traffic events in region r, we use their natural language description, and by employing the GloVe pre-trained distributed word vectors [27], we create a description to vector (Desc2Vec) representation for r. Such representation is the average representation of words in description of all events which took place within r during a particular time period. Size of this vector is 100. The choice of GloVe among the existing models is because of its well-known applicability for generic applications and also reasonable dictionary size (i.e., 400K terms). We obtain traffic events from [22].

In this way, we represent r during time interval t by 24 time-variant (i.e., traffic, time, and weather) and 113 time-invariant (i.e., POI and Desc2Vec) attributes. In order to predict the label of r during t, we

[7] These time intervals are [6am – 10am], [10am – 3pm], [3pm – 7pm], [7pm – 10pm], and [10pm – 6am].

use a vector representing the last 8 time intervals (last two hours), including one instance of time-invariant attributes (113 features) and 8 instances of time-variant attributes (8×24 features)[8].

5.2 Deep Accident Prediction (DAP) Model

To better utilize heterogeneous sources of data and perform real-time traffic accident prediction, we propose a deep neural network model, named the Deep Accident Prediction (DAP). This model is shown in Figure 4, and we describe its components as follows.

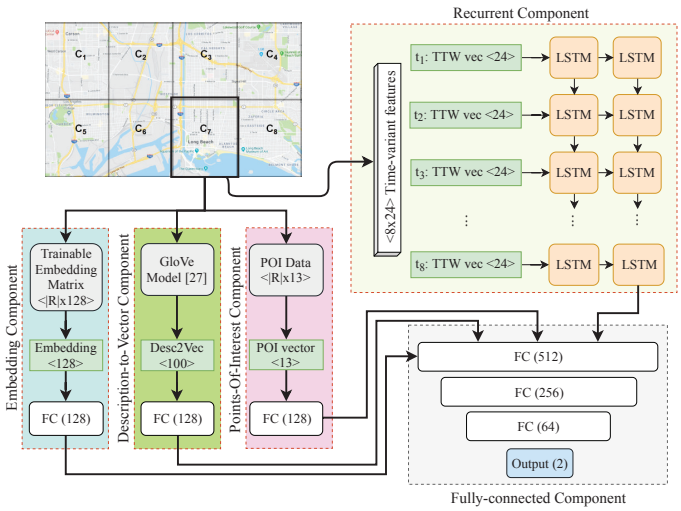


Figure 4: DAP: A Deep neural-network-based Accident Prediction model. Here, R is the set of all regions; each C_i is a grid-cell (or region); and FC, TTW, and POI stand for fully connected, time-traffic-weather, and point-of-interest, respectively.

- **Recurrent Component**: Regarding the definition of our prediction framework, we use a set of 8 vectors, each of size 24 (i.e., time-variant attributes), which can be treated as a sequence of such vectors (given their temporal order); therefore, we may benefit from the recurrent neural network models. Specifically, we use a long-short-term-memory (LSTM) model [12] as represented in Figure 4, which includes two recurrent layers, each with 128 LSTM cells. Thus, the output is a vector of size 128.
- **Embedding Component**: Given the index of a grid-cell, this component provides a distributed representation of that cell which encodes essential information in terms of spatial heterogeneity, traffic characteristics, and impact of other environmental stimuli on accident occurrence. This distributed representation will be derived as we train the entire pipeline. We feed this representation to a feed-forward layer of size 128 that uses the *sigmoid* activation function. Note that the embedding matrix is of size $|R| \times 128$, where R is the set of all grid-cell regions in input dataset.
- **Description-to-Vector Component**: This component utilizes the natural language description of historical traffic events in a grid-cell, that is, Desc2Vec data. We feed Desc2Vec of a grid-cell to a feed-forward layer of size 128 using the *sigmoid* activation function.
- **Points-of-Interest Component**: This component utilizes points-of-interest data (a vector of size 13), which is a representation of spatial characteristics. We feed a POI vector to a feed-forward layer of size 128 which also uses the *sigmoid* activation function.
- **Fully-connected Component**: This component utilizes the output of above components to make the final prediction. Here we have four dense layers of size 512, 256, 64, and 2, respectively. Additionally, to speed-up the training process, we use batch normalization [14] after the second and the third layer. We use ReLU as the activation function of the first three layers, and apply *softmax* on the output of the last layer.

The DAP model utilizes inputs of various types to better capture temporal and spatial heterogeneity. Using DAP we are able to extract latent spatio-temporal features in terms of embedding representations, whose impact we show through our real-world experiments. We employed grid-search to perform hyper-parameter tuning to find the optimal number of recurrent layers (choices of $\{1, 2, 3\}$); the best type of recurrent cells (choices of $\{Vanilla\text{-}RNN, GRU, LSTM\}$); size of the embedding vector for grid-cells (choices of $\{50, 100, 150\}$); sizes of the different fully connected layers (choices of $\{64, 128, 256, 512\}$); and activation function for each fully connected layer (choices of $\{sigmoid, ReLU, tanh\}$). We employed the Adam optimizer [16] with an initial learning rate of 0.01 to train the model.

6 EXPERIMENTS AND RESULTS

In this section we first describe the data which is used for prediction and analysis. Then, we describe baseline models. Next we compare different models using a variety of metrics, followed by analyses of data attributes. All implementations are in Python using Tensorflow [11], Keras [9], and scikit-learn [26] libraries; and experiments were run on nodes at the Ohio Supercomputer Center [4] [9].

6.1 Data Description

To evaluate our accident prediction framework, we chose six cities: *Atlanta*, *Austin*, *Charlotte*, *Dallas*, *Houston*, and *Los Angeles*; primarily so as to achieve diversity in traffic and weather conditions, population, population density, and urban characteristics (road-network, prevalence of urban versus highway roads, etc.). We sampled a subset of data (traffic, weather, etc.) collected from June 2018 to August 2018 (i.e., 12 weeks) for each city. We chose this time period to prevent any noises as result of seasonality in weather and traffic patterns. To create Desc2Vec for each grid cell region, we used traffic events which took place within that region from June 2017 to May 2018 (i.e., a one-year time frame), where data obtained from the *Large-Scale Traffic and Weather Events dataset* [22]. From the traffic, time, weather, POI, and Desc2Vec data for each grid cell, and by scanning through the data with a window of size 2 hours and 15 minutes and a shift of 15 minutes (see Figure 5), we built a *sample entry* using data of the first two hours (see Section 5.1). Each entry is represented by 113 time-invariant and 8×24 time-variant features. The last 15 minutes is used to label the sample entry as an accident or non-accident case.

Since accidents are rare and because our dataset is sparse [10], we performed *negative sampling* to balance the frequency of samples between accident and non-accident classes. Specifically, we uniformly sampled from the non-accident class with a probability of 2.0%. Table 5 summarizes the number of samples for each class (Acc

[8] See Section 3 for formulation of prediction task.

[9] Code and sample data is available at https://github.com/mhsamavatian/DAP.

[10] Our data is result of streaming data with possibility of missing records.

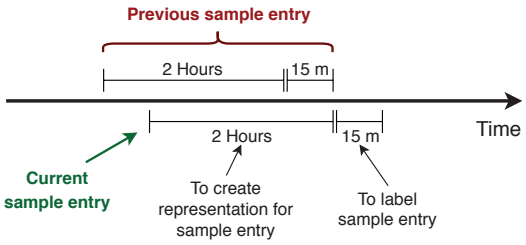


Figure 5: Creating a Sample Entry (see Section 6.1).

versus Non-Acc), for each city, after negative sampling. As can be seen, the maximum ratio of accident to non-accident is about 27% for *Los Angeles* (which is still lower than the ratio which is employed by previous studies; e.g., Yuan et al. [37] employed 33%). Table 5 also shows the number of all other traffic events (except accidents) which took place during the selected 12 weeks time frame. We use data from the first 10 weeks to train and data from the last two weeks as the test set, for each city.

Table 5: Distribution of accident (Acc) and non-accident (Non-Acc) classes, and traffic events (except accidents).

City	#Acc	#Non-Acc	Acc/Non-Acc	#Traffic Events
Atlanta (GA)	2,630	11,970	22%	24,396
Austin (TX)	4,274	23,280	18%	16,313
Charlotte (NC)	5,295	20,192	26%	14,030
Dallas (TX)	3,363	28,537	12%	28,098
Houston (TX)	5,859	43,762	13%	40,735
Los Angeles (CA)	7,974	29,020	27%	97,090

6.2 Baseline Models

We chose logistic regression (LR), gradient boosting classifier (GBC), and a Deep Neural Network (DNN) model as baselines.

- **Logistic Regression (LR)**: A significant number of previous studies leveraged regression-based models to perform accident prediction [5, 7, 8]. Therefore, we employ logistic regression as a reasonable baseline to perform our binary classification task.
- **Gradient Boosting Classifier (GBC)**: GBC is a popular general-purpose classification model, with useful boosting characteristics and a suitable learning process. In practice, GBC usually provides superior results for binary or multi-class classification tasks, when compared to the other models such as Random Forest or Support Vector Machine; our preliminary experiments also confirmed this.
- **Deep Neural Network (DNN)**: This is a four-layer feed-forward neural network, with three hidden layers of size 512, 256, and 64, respectively. *ReLU* was used as the activation function of the hidden layers, and *softmax* was applied on the output of the last layer. To speed-up the training process, we used batch normalization [14] after the second and third hidden layers. We employed the Adam optimizer [16] with an initial learning rate of 0.01 to train this model.

As input, the baseline models utilize vectors of size 305, that includes 113 time-invariant and 192 time-variant attributes (see Section 5.1). The output is the prediction probability for "accident" and "non-accident" classes. Using grid-search over heuristic choices of parameters, we found the best parameter setting for each model. For LR, we performed the grid search over choices of regularizations: $\{L1, L2\}$, maximum iterations: $\{100, 100, 10000, 100000\}$, and solvers: $\{newton\text{-}cg, lbfgs, sag, liblinear\}$. For GBC, the grid search was performed over choices of learning rates: $\{0.01, 0.05, 0.1, 0.15\}$, number of estimators: $\{100, 200, 300, 400\}$, and maximum depth: $\{3, 4, 5, 6\}$. For DNN, the grid search was performed over choices of initial learning rates: $\{0.001, 0.01, 0.05, 0.1\}$, activation functions: $\{sigmoid, ReLU\}$, number of hidden layers: $\{2, 3, 4\}$, and size of hidden layers: $\{128, 256, 512\}$.

6.3 Exploring Models

In this section we evaluate different models based on their ability to predict traffic accidents. That is, we compare different models based on *F1-score* (defined by Equation 2), reported for each class separately, as well as the *weighted average F1-score* (the relative frequency of each class is used as its weight).

$$\begin{aligned} Precision &= \frac{\text{true positive}}{\text{true positive + false positive}} \\ Recall &= \frac{\text{true positive}}{\text{true positive + false negative}} \\ F1\text{-}Score &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision + Recall}} \end{aligned} \tag{2}$$

We used logistic regression (LR), gradient boosting classifier (GBC), and a deep neural network (DNN) model as baselines. We report the result of our DAP model, as well as a variation of DAP without the embedding component (DAP-NoEmbed). We ran each model three times, and reported the average results. As mentioned before, we used grid search to find the optimal parameters. For LR and GBC, we performed this for each city, but for the neural-network-based models we employed grid search for one city and used the best architecture setting for the other cities. DNN, DAP, and DAP-NoEmbed were trained for 60 epochs, and using *early stopping* based on the validation set (i.e., 10% of the training set), we used the best model for prediction on the test set. It is worth noting that each model is separately trained and tested for each city and we do not train a single model for all cities. Table 6 presents the results of this experiment. In this table we report *F1-score* for class of accident (Acc), non-accident (Non-Acc), and the weighted average (W-Avg). We note that the class of accident is usually more important, while we seek to provide reasonable results for the other class (non-accident) as well. LR and GBC usually provide better results for non-accident class, and given the frequency of this class, their weighted average score is also reasonably high. However, when considering the accident class, we note that neural-network-based models provide more satisfactory results, where our proposed DAP model provides superior results for 5 of the 6 cities (DNN provided the best result for Houston). Considering the weighted average on *F1-score*, we note that DAP provides better results when compared to the other neural-network-based models.

To better compare different models, Figure 6 shows the average results of different models across all six cities, by separately reporting *F1-score* for class of accident and non-accident, and the weighted average *F1-score*. As one can see, our proposed model provides a significant improvement for class of accidents, while LR and GBC provide slightly better results for the non-accident class. When considering the weighted average, we observe LR, DAP and GBC slightly outperform the other models. Once again note that the "accident class" is the one of most importance, given that accidents are rare events. Hence we should pay more attention to false negatives (i.e., predicting an accident as a non-accident) rather than false positives (i.e., predicting a non-accident as an accident).

Table 6: Accident prediction results based on F1-score for class of accidents (Acc), non-accidents (Non-Acc), and weighted average (W-avg).

Model / City	LR			GBC			DNN			DAP-NoEmbed			DAP		
	Acc	Non-Acc	W-Avg	Acc	Non-Acc	W-Avg	Acc	Non-Acc	W-Avg	Acc	Non-Acc	W-Avg	Acc	Non-Acc	W-Avg
Atlanta	0.54	0.91	0.83	0.57	0.91	0.84	0.62	0.89	0.83	0.62	0.91	0.84	**0.65**	0.89	0.84
Austin	0.58	0.93	0.87	0.61	0.93	0.87	0.62	0.92	0.87	0.62	0.93	0.87	**0.64**	0.91	0.87
Charlotte	0.56	0.91	0.83	0.60	0.91	0.84	0.61	0.87	0.82	0.61	0.87	0.81	**0.63**	0.87	0.82
Dallas	0.30	0.94	0.87	0.32	0.94	0.87	0.36	0.94	0.87	0.43	0.88	0.83	**0.50**	0.93	0.88
Houston	0.49	0.94	0.88	0.51	0.94	0.88	**0.59**	0.93	0.88	0.58	0.92	0.88	0.58	0.93	0.88
Los Angeles	0.41	0.88	0.78	0.45	0.88	0.79	0.53	0.81	0.75	0.53	0.77	0.72	**0.56**	0.84	0.78

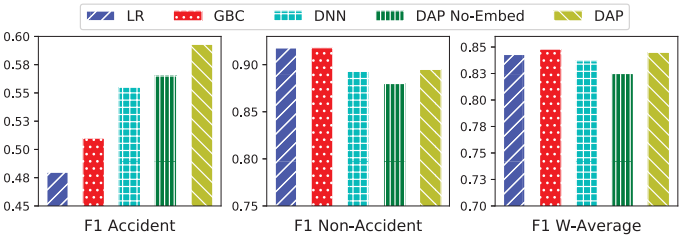


Figure 6: Comparing different models based on average $F1\text{-}score$ (across all six cities) for class of accident, non-accident, and weighted average.

While we cannot directly compare our proposal with the state-of-the-art models such as [8, 18, 37] (due to inconsistency between input types, unavailability of input data used by those models, inconsistency between reported metrics, etc.), we note that their reported results based on $F1\text{-}score$ show similar trend and values (see [37] for example). Further, we believe that separately reporting prediction results for different classes (i.e., accident versus non-accident) provides a better context to compare different solutions.

6.4 Exploring Features

Our next experiment was to examine the importance of different feature categories for the task of accident prediction. For this exploration, we designed two testing scenarios as follows:

- **Only One**: This scenario means we only use one category of features (traffic, POI, time, etc.) to perform accident prediction.
- **All But One**: This scenario means to remove only one category of features and perform the prediction task.

For this experiment, we only report the result of GBC and DAP-NoEmbed, and omit the results of other models for the interest of space. Also, because of having the trainable embedding component, we choose DAP-NoEmbed over DAP to exclude the effect of the embedding component when studying the impact of other features[11]. Figure 7 demonstrates the results, where we report weighted average $F1\text{-}score$, and $F1\text{-}score$ on accident class. Based on parts (a), (b), (e), and (f), we generally observe weather (WE) and time (TM) are the least important categories of attributes to be used alone[12]. However, parts (c), (d), (g), and (h) reveal that removing time attributes would significantly hurt the prediction performance. Based on these figures, when we remove Desc2Vec, POI, and Traffic attributes (i.e., D+P+TR), the prediction performance drops significantly, which shows the importance of these categories. We may also note that these categories might have correlation, where removing one of them does not significantly change the prediction results (see (c) and (d)). Therefore, when we remove all three, then we observe a significant drop.

It is worth noting that among the POI types, we found "crossing", "junction", "stop", and "traffic signal" to be more effective than the others for the task of accident prediction.

[11]Since DAP utilizes an embedding component, we cannot fairly study the impact of several categories of features (in isolation), such as traffic, weather, and points-of-interest; given the correlation between these categories and the latent representation which will be derived for each region.

[12]Note that we could not use categories D2V and POI for DAP-NoEmbed, regarding the architecture of this model.

7 CONCLUSION AND FUTURE WORK

Traffic accidents are a major public safety issue, with much research devoted to analysis and prediction of these rare events. However, most of the studies suffer from using small-scale datasets, relying on extensive data that is not easily accessible to other researchers, and being not applicable for real-time purposes. To address these challenges, we introduced a new framework for real-time traffic accident prediction based on easy-to-obtain, but sparse data. Our prediction model incorporated several neural network based components that used a variety of data attributes such as traffic events, weather data, points-of-interest, and time information. We also created a publicly available countrywide traffic accident dataset, named US-Accidents, through a comprehensive process of data collection, cleansing, and augmentation. Using the data from US-Accidents, we compared our work against several neural-network-based and traditional machine leaning models, and showed its superiority by means of extensive experiments. Further, we studied the impact of different categories of data attributes for traffic accident prediction, and found time, traffic events, and points-of-interest as having significant value. In the future, we plan to incorporate other publicly available sources of data (e.g., demographic information and annual traffic reports) for the task of real-time traffic accident prediction.

ACKNOWLEDGMENTS

This work is supported by a grant from the NSF (EAR-1520870), one from the Nationwide Mutual Insurance (GRT00053368), and another from the Ohio Supercomputer Center (PAS0536). Any findings and opinions are those of the authors.

REFERENCES

[1] Joaquín Abellán, Griselda López, and Juan De OñA. 2013. Analysis of traffic accident severity using decision rules via decision trees. *Expert Systems with Applications* 40, 15 (2013), 6047–6054.
[2] Bing Map Traffic API. 2019. https://www.bingmapsportal.com/. (2019). Accessed: 2019-09-1.
[3] Ciro Caliendo, Maurizio Guida, and Alessandra Parisi. 2007. A crash-prediction model for multilane roads. *Accident Analysis & Prevention* 39, 4 (2007), 657–670.
[4] Ohio Supercomputer Center. 1987. Ohio Supercomputer Center. (1987). http://osc.edu/ark:/19495/f5s1ph73

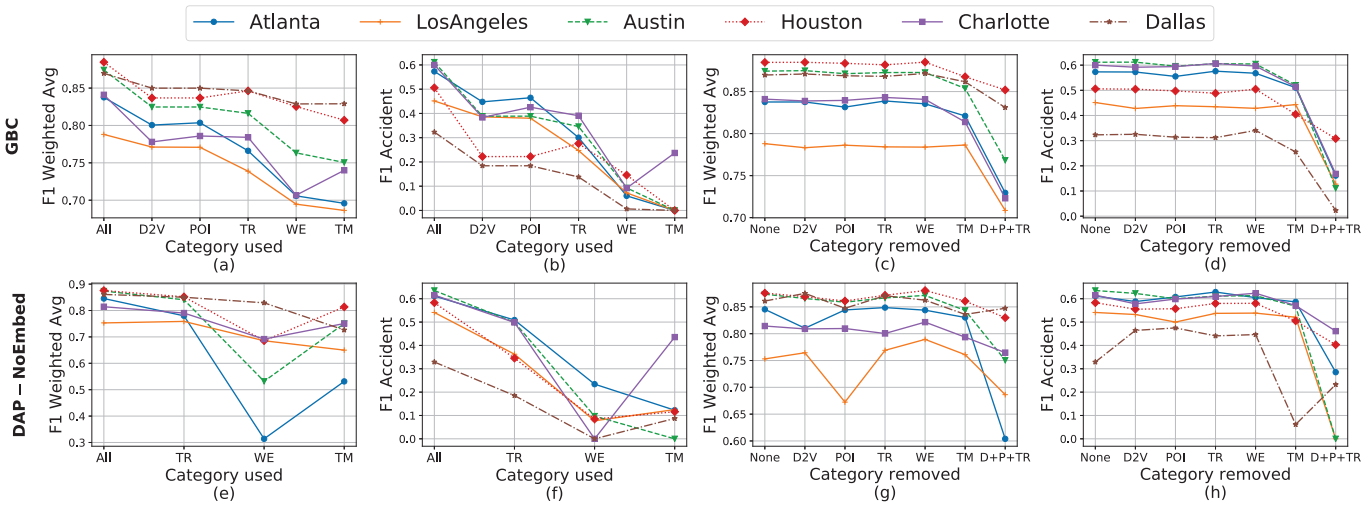


Figure 7: Prediction results using only one category (a, b, e, and f), and all but one category (c, d, g, and h). Here D2V, TR, WE, and TM stand for Desc2Vec, traffic, weather, and time, respectively. Also, "D+P+TR" means removing Desc2Vec, POI, and traffic from input features.

[5] Li-Yen Chang. 2005. Analysis of freeway accident frequencies: negative binomial regression versus artificial neural network. *Safety science* 43, 8 (2005), 541–557.

[6] Li-Yen Chang and Wen-Chieh Chen. 2005. Data mining of tree-based models to analyze freeway accident frequency. *Journal of safety research* 36, 4 (2005), 365–375.

[7] Chao Chen, Xiaoliang Fan, Chuanpan Zheng, Lujing Xiao, Ming Cheng, and Cheng Wang. 2018. SDCAE: Stack Denoising Convolutional Autoencoder Model for Accident Risk Prediction Via Traffic Big Data. In *2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)*. IEEE, 328–333.

[8] Quanjun Chen, Xuan Song, Harutoshi Yamada, and Ryosuke Shibasaki. 2016. Learning deep representation from big and heterogeneous data for traffic accident inference. In *Thirtieth AAAI Conference on Artificial Intelligence*. AAAI, Palo Alto, CA, USA.

[9] François Chollet et al. 2015. Keras. https://github.com/fchollet/keras. (2015).

[10] Daniel Eisenberg. 2004. The mixed effects of precipitation on traffic crashes. *Accident analysis & prevention* 36, 4 (2004), 637–647.

[11] Abadi et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). https://www.tensorflow.org/ Software available from tensorflow.org.

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[13] Chukwutoo C Ihueze and Uchendu O Onwurah. 2018. Road traffic accidents prediction modelling: An analysis of Anambra State, Nigeria. *Accident Analysis & Prevention* 112 (2018), 21–29.

[14] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[15] 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.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Sachin Kumar and Durga Toshniwal. 2015. A data mining framework to analyze road accident data. *Journal of Big Data* 2, 1 (2015), 26.

[18] Lei Lin, Qian Wang, and Adel W Sadek. 2015. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. *Transportation Research Part C: Emerging Technologies* 55 (2015), 444–459.

[19] Fred L Mannering, Venky Shankar, and Chandra R Bhat. 2016. Unobserved heterogeneity and the statistical analysis of highway accident data. *Analytic methods in accident research* 11 (2016), 1–16.

[20] MapQuest Traffic API. 2019. https://www.mapquest.com/. (2019). Accessed: 2019-09-1.

[21] Sobhan Moosavi, Behrooz Omidvar-Tehrani, R Bruce Craig, Arnab Nandi, and Rajiv Ramnath. 2017. Characterizing driving context from driver behavior. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, New York, NY, USA, 46:1–46:4. https://doi.org/10.1145/3139958.3139992

[22] 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 *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, New York, NY, USA, 2905–2913. https://doi.org/10.1145/3292500.3330755

[23] Alameen Najjar, ShunâĂŹichi Kaneko, and Yoshikazu Miyanaga. 2017. Combining satellite imagery and open data to map road safety. In *Thirty-First AAAI Conference on Artificial Intelligence*. AAAI, Palo Alto, CA, USA.

[24] Nominatim Tool. 2019. https://wiki.openstreetmap.org/wiki/Nominatim. (2019). Accessed: 2019-09-1.

[25] Open Street Map (OSM). 2019. https://www.openstreetmap.org/. (2019). Accessed: 2019-09-1.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[28] World Health publisher. 2015. *Global status report on road safety 2015*. World Health publisher.

[29] Honglei Ren, You Song, Jingwen Wang, Yucheng Hu, and Jinzhi Lei. 2018. A Deep Learning Approach to the Citywide Traffic Accident Risk Prediction. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 3346–3351.

[30] JD Tamerius, X Zhou, R Mantilla, and T Greenfield-Huitt. 2016. Precipitation effects on motor vehicle crashes vary by space, time, and environmental conditions. *Weather, Climate, and Society* 8, 4 (2016), 399–407.

[31] 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.

[32] Time And Date website. 2019. https://www.timeanddate.com/. (2019). Accessed: 2019-09-1.

[33] Traffic Message Channel (TMC) Code. 2019. https://wiki.openstreetmap.org/wiki/TMC/Event_Code_List. (2019). Accessed: 2019-09-1.

[34] Weather Underground. 2014-2019. https://www.wunderground.com/. (2014-2019). Accessed: 2019-09-1.

[35] Lu Wenqi, Luo Dongyu, and Yan Menghua. 2017. A model of traffic accident prediction based on convolutional neural network. In *2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE, 198–202.

[36] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. 2018. Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, New York, NY, USA, 984–992.

[37] Zhuoning Yuan, Xun Zhou, Tianbao Yang, James Tamerius, and Ricardo Mantilla. 2017. Predicting traffic accidents through heterogeneous urban data: A case study. In *Proceedings of the 6th International Workshop on Urban Computing (UrbComp 2017), Halifax, NS, Canada*, Vol. 14. ACM, New York, NY, USA.