Hetero-ConvLSTM: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data

Zhuoning Yuan, Xun Zhou, Tianbao Yang
University of Iowa
Iowa City, IA 52242, USA
{zhuoning-yuan,xun-zhou,tianbao-yang}@uiowa.edu

ABSTRACT

Predicting traffic accidents is a crucial problem to improving transportation and public safety as well as safe routing. The problem is also challenging due to the rareness of accidents in space and time and spatial heterogeneity of the environment (e.g., urban vs. rural). Most previous research on traffic accident prediction conducted by domain researchers simply applied classical prediction models on limited data without addressing the above challenges properly, thus leading to unsatisfactory performance. A small number of recent works have attempted to use deep learning for traffic accident prediction. However, they either ignore time information or use only data from a small and homogeneous study area (a city), without handling spatial heterogeneity and temporal auto-correlation properly at the same time.

In this paper we perform a comprehensive study on the traffic accident prediction problem using the Convolutional Long Short-Term Memory (ConvLSTM) neural network model. A number of detailed features such as weather, environment, road condition, and traffic volume are extracted from big datasets over the state of Iowa across 8 years. To address the spatial heterogeneity challenge in the data, we propose a Hetero-ConvLSTM framework, where a few novel ideas are implemented on top of the basic ConvLSTM model, such as incorporating spatial graph features and spatial model ensemble. Extensive experiments on the 8-year data over the entire state of Iowa show that the proposed framework makes reasonably accurate predictions and significantly improves the prediction accuracy over baseline approaches.

CCS CONCEPTS

• **Information systems → Geographic information systems**; **Data mining**; *Data management systems*;

KEYWORDS

Traffic accident prediction; Deep learning; Spatial heterogeneity; Convolutional LSTM

ACM Reference Format:
Zhuoning Yuan, Xun Zhou, Tianbao Yang. 2018. Hetero-ConvLSTM: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3219819.3219922

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219922

1 INTRODUCTION

Traffic accidents have been one of the most significant public safety issues. According to the World Health Organization (WHO), more than 1.25 million people die each year as a result of road traffic accidents [29]. Road traffic injuries are the leading causes of death among young population between 15 and 29. Reducing traffic accident is a crucial societal problem. The ability to understand and forecast potential accidents in the future (e.g., where, when, or how) is thus very useful not only to public safety stakeholders (e.g., police) but also to transportation administrators and individual travelers.

Numerous research have shown that environmental attributes such as weather, road conditions, and light condition might have an impact on the risk of traffic accident [7, 15, 28]. With the rapid development of data collection techniques and the availability of big datasets in recent years, abundant environmental data, public transportation records, and motor vehicle crash reports can be collected and fused, which makes predicting traffic accidents more realistic.

However, traffic accident prediction is a very challenging problem. First of all, the causes of traffic accidents are complex. Besides the common factors listed above, random factors such as vehicle mechanical problems, driver carelessness may also cause traffic accidents. Second, traffic accidents are rare events. Precisely predicting individual accidents is challenging due to lack of enough samples. Finally, the factors that may cause traffic accidents vary from place to place. For example, the main factors that lead to traffic accidents in an urban region with busy local roads might be very different from on a rural express way. Handling the spatial heterogeneity in the data is challenging.

Commonly, the traffic accident prediction problem has been formulated as a classification problem or a regression problem. For example, some work aim to predict whether or not an accident will occur at a specific location or in a specific area (e.g., road segment) during each time window (e.g., hour, day) [9, 10, 12, 17]. Other work [7, 8, 15, 23] predict the number of accidents at given time and locations using regression models. These work, however, typically use classical data mining methods and do not consider the unique features of traffic accident data such as spatial heterogeneity and temporal auto-correlation, leading to unsatisfactory performance. A limited number of recent works have made attempts to solve the problem using deep learning approaches, such as deep neural network or convolutional neural networks [11, 19, 26]. In this study, the problem is formulated as an image prediction problem, where a traffic risk map is generated by learning from the traffic accident records and other data (e.g., satellite image, cell phone records). These works, either ignore the temporal information (e.g., predict a static risk map), or performed the analysis on datasets at small spatial scales with very limited features, therefore lacking the ability to forecast accidents at larger scale (e.g., in a state) with spatial heterogeneity (e.g., both urban and rural areas).

In this paper we propose Hetero-ConvLSTM, a deep learning approach for traffic accident prediction based on big heterogeneous spatio-temporal data. The entire study area is partitioned into grid cells. A number of fine-grained urban and environmental features such as traffic volume, road condition, rainfall, temperature, and satellite images are collected and map-matched with each grid cell. Given the number of accidents as well as the other urban and environmental features at each location, we learn a model to predict the number of accidents that will occur in each grid cell in future time slots. We adopt the Convolutional Long Short-Term Memory (ConvLSTM) neural network model and incorporate spatial features in the model to better capture the temporal trends and spatial heterogeneity of the data. In addition, we propose a model ensemble framework, where different models are learned for different regions of the study area and the results are assembled to generate the final prediction. **To the best of our knowledge, this is the first work that address spatial heterogeneity in the framework of deep neural networks for traffic accident prediction**. Also it is the first work that uses ConvLSTM for traffic accident prediction. Results show that our model outperforms classical models and achieve higher accuracy.

We highlight our contributions as follows:

- We collect and fuse heterogeneous big datasets including road, weather, time, traffic, and human factors for traffic accident prediction. This, to the best of our knowledge, has not been done in prior research
- For the first time in the literature of traffic accident prediction, we incorporate the spatial structure of the road network into the predictive models by leveraging new features generated through eigen-analysis of the road network to address the spatial heterogeneity challenge
- We propose a deep learning framework using the Convolutional Long Short-Term Memory (LSTM) Neural Network with a model ensemble approach to further address the spatial heterogeneity in the data and improve the accuracy of the prediction
- We perform comprehensive experiments on various parameter settings, feature sets, and baseline approaches. We discover from the results that the major factors for traffic accident change over space. For rural areas weather and SpatialGraph features play important roles and in urban areas road condition, traffic volume, and holiday/weekday information are more important

The rest of the paper is organized as follows: Section 2 discusses the related work of this paper. Section 3 presents our datasets and the problem formulation. Section 4 introduces our feature extraction steps. Section 5 presents our Hetero-ConvLSTM approach. Section 6 presents case studies and experimental results. Section 7 discusses the challenges of deploying our framework. Section 8 concludes the entire paper.

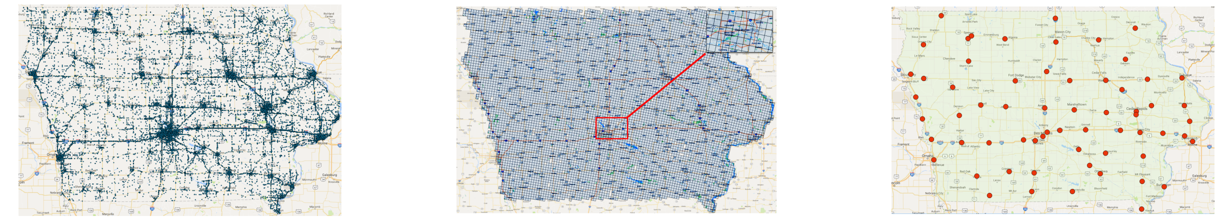
2 RELATED WORK

Traffic Accident Prediction using Classical Techniques: A big body of literature from public safety and injury prevention researchers aim to classify each given road segment at given time into binary classes {Accident, No Accident}. Chang [9] compared the performance of Artificial Neural Network with that of a negative binomial regression model over 1338 accidents. ANN achieved 64% and 61.4% accuracy for training and testing, respectively. Chang et al. [10] also applied the decision tree model on the same dataset to predict highway accidents. The training and testing accuracy are less than 55%. Olutayo et al. [12] applied decision tree and ANN model on a dataset from Nigeria and achieved precision and recall both around 0.52. Lin et al. [17] employed FP-Tree to select features that are more likely to contribute to the prediction. Then they applied Random Forest, K-Nearest Neighbor, and Bayesian Network to predict accidents along the same road. The best performance archived is around 61%. Abellán et al. [5] used a Probabilistic Neural Network (PNN) model to predict traffic accident based on real-time road condition (e.g., traffic volume, speed).

Some other work aim at fitting regression or other models to predict the number of traffic accidents on specific roads or in certain regions. Many of them try to identify correlations between attributes (e.g., weather, road conditions) and the accident risk. Caliendo et al. [8] developed Poisson, Negative Binomial, and Negative Multinomial regression models to predict the number of accidents on given roads. Oh et al. [23] employed a zero-inflated Poisson regression model to predict the number of crashes at railway-highway intersections. They identified the correlation between a number of factors and crash rate. Bergel-Hayat et al. [7] employed an auto-regressive regression model to study the correlations between weather attributes and injury accidents. Eisenberg et al. [15] used negative binomial regression model to study the relationship between monthly precipitation and monthly fatal crashes. Tamerius et al. [28] analyzed the Relative Accident Rate (RAR) to study the relationship between precipitation and motor vehicle crashes over space and time.

All of the above work simply apply classical data mining techniques on small scale traffic accident data (e.g., one or a small number of roads) with limited features. Also they typically don't address unique data properties such as time periodicity, spatial auto-correlation and heterogeneity, therefore having relatively low accuracy.

Deep Learning Models for Traffic Accident Prediction: Some recent work have attempted to tackle the traffic accident analysis problem using deep learning models. Chen et al. [11] utilized mobile phone data and historical accident records to build a model for real-time accident risk assessment. Najjar et al. [19] used Convolutional Neural Netowrk (CNN) to predict traffic accident risk map using historical accident data and satellite images. Both of the above works, however, only predict a time-invariant accident risk map. This is not suitable for real-time accident prediction for safety planning. A recent work by Ren et al. [26] used Long Short-Term Memory (LSTM) model to predict traffic accident risk in the near



(a) Visualization of Traffic Accidents **(b) Rainfall Map** **(c) RWIS Observation Stations**

Figure 1: Illustrations of the Motor Vehicle Crash (2013), Rainfall, and RWIS datasets.

future. However, this work is a simple application of the model on traffic accident data, without addressing spatial heterogeneity of the data. The study area is also limited to urban region (Beijing). By contrast, our paper proposes a few novel ideas to address the spatial heterogeneity issues in the data we collect. We implement these ideas on top of a LSTM neural network. To the best of our knowledge, this is the first work using deep learning model to address spatial heterogeneity in traffic accident forecasting.

Traffic Accident Hotspot Detection: Other work on traffic accident analysis include clustering and hotspot detection. Shi et al. [27] proposed a likelihood-ratio test approach to identify road intersections with high traffic accident density. Oliver et al. [24] proposed a linear hotspot detection approach to identify paths with significantly higher density of traffic accidents compared to others. Wang et al. [31] proposed an ontology-based approach, which considers the severity level of accidents for traffic accident risk clustering and mapping. These work are unsupervised analysis thus not directly related to our paper.

3 OVERVIEW

This section presents the data we collected and introduces the formulation of our problem.

3.1 Data Sources

We choose the state of Iowa, United State, as the study area. Iowa is a state with both rural and urban environments, and various extreme weather conditions (e.g., snow storm, heavy rains, tornadoes). Tamerius et al. [28] pointed out that Iowa is ideal for studying the impact of precipitation on traffic accidents due to its varying weather conditions. All the data we collected are about Iowa, as detailed below.

Motor Vehicle Crash Data. We obtained motor vehicle crash data from the Iowa Department of Transportation (DOT) [14]. The data contains the crash records from 2006 to 2013. In addition to the basic information, i.e., time and location of a traffic accident, the dataset also contains many valuable features related to the accident such as road information. Figure 1(a) shows the mapping of the crash locations in Iowa on top of the major highway network in 2013.

High-Resolution Rainfall Data. We also obtained Stage IV radar-rainfall product developed by the NWS [18]. The data contains hourly precipitation amount (in millimeter) from radar at 4 kilometer resolution. There are totally 8,026 observation tiles, which cover the entire Iowa over the study time periods. Figure 1(b) shows the map of the observation tiles.

RWIS (Roadway Weather Information System) Data. RWIS is a project of monitoring the temperature change, maintained by Iowa Department of Transportation (DOT) [22] . It contains 86 observation stations that are located near the primary roads of Iowa, e.g., there are 14 stations along with Interstate-80. The project mainly provides temperature and wind related features. We collect the data from 2006 to 2013. Figure 1(c) shows the locations of the observation stations.

Road Networks. We collected three different road network datasets from Iowa DOT GIS data portal [21] with basic road information in the state of Iowa, detailed speed limits of the road, and the most recent estimated Annual Average Daily Traffic (AADT) volume for the primary and secondary roads. The AADT data also include detailed statistics of each type of vehicles, such as Single Unit Truck AADT and Combination Truck AADT for every road.

Satellite Images. We also collect a satellite image of Iowa from Google Earth [16].

Traffic Camera Data. We collect real-time traffic volume data from a total of 128 camera stations along major highways in Iowa from 2006 to 2013. The number of vehicles along both directions in each hour are recorded at these locations.

3.2 Problem Formulation

We impose a spatial grid S on the study area, where each grid s_i represents a $d \times d$ square region. For example, if $d = 5km$, the entire state of Iowa can be partitioned into 128×64 grids.

In this problem, we aim to learn a model to predict the total number of accidents in each grid in S during each time slot. The time slot could be, for example, an hour, a day, or a week. In this paper, we choose each day (24 hours) as the length of t. However, one can apply our proposed framework with different choice of d, and t. Note time-variant features such as weather, traffic volume of day t are not available until day t has passed. So we only use features for days up to $t - 1$ to predict the accidents on day t.

Specifically, we formulate the problem as follows:

- **Given:**
  - A spatio-temporal field $S \times T$, where $S = \{s_1, s_2, ..., s_n\}$ is a spatial grid, and $T = \{t_1, t_2, ..., t_n\}$ is the time length of the study period partitioned into equal-length slots
  - A 3-D tensor of traffic accident count C, where $C(s, t)$ is the number of accidents in grid $s \in S$ during day $t \in T$
  - A list of m feature tensors $F = \{f_1, f_2, ..., f_m\}$, where f_k is a three-dimensional tensor recording the corresponding attribute in each grid s_i during each time slot t

- A training data set $D_{train} = \{C(S,t), F(S,t)\}$ where $t \in T_{train}$, and a testing data set $D_{test} = \{C(S,t), F(S,t)\}$ where $t \in T_{test}$
- **Find:**
  - A model to predict $C(S,t)$ for every $t \in T_{test}$
- **Objective:**
  - Minimize prediction error
- **Constraints:**
  - The dependency between C and F vary spatially
  - All accidents occur along the road network
  - $\mathrm{F}(\mathrm{S}, t_i)$ is not available for prediction of $\mathrm{C}(\mathrm{S}, t_i)$, $t_i \in T_{test}$

4 FEATURE EXTRACTION

To generate the features for our input, we match the collected datasets with each (s_i, t) combination and aggregate the data to extract a list of features.

Dependent Variable Tensor (C): For each grid s_i in each day t, we count the total number of accidents ($C(s_i, t)$). We totally matched 375,690 motor vehicle crashes over 8 years.

The extraction of independent features F are detailed below.

4.1 Time-Invariant Features

Road Network Mask Feature Map (N): We map the road network (with primary and secondary roads) onto the grids and create a mask layer. Although the entire study area is partitioned into grid cells, it is obvious that traffic accidents can only occur on the road network. To make sure that the prediction results make sense, we convert the road network into a feature map. Since the road network merely changes over years, this feature is time-invariant. Specifically, for each grid s_i, we define the value of the feature $N(s_i, t)$,

$$N(s_i, t) = \begin{cases} 1 & \text{if there are roads in } s_i \\ 0 & \text{otherwise} \end{cases}$$

Road Condition Features (RC): In addition to the network mask layer, we calculate the average length of all the roads and the average speed limit of the roads in each grid cell and store each of these measures as a feature. We also include the features relevant to road property, such as, number of intersections, number of lanes, road function, road curve and Annual Average Daily Traffic(AADT). These features are time invariant. We obtain six features for each grid s_i.

Google Earth Satellite Image (G): We obtain a snapshot satellite image of the entire Iowa from Google Earth [16] and geo-register the image to the map of Iowa based on the latitudes and longitudes of the corners of the image. The image is decomposed to three color channels (R, G, B), each as a feature. For each grid cell s_i for each channel, the feature value is the average value of the pixels that overlap with s_i. These three features are time invariant.

4.2 Time-Variant Features

Rainfall Feature (RA): We map each radar data tile from the rainfall dataset [18] onto the grid cell s_i in our framework. For each grid cell s_i, we find all the rainfall radar data tiles r_j that overlap with s_i. The rainfall amount $RA(s_i, t)$ is calculated as the average daily total rainfall of all such tiles r_j on day t. This result in one time-variant feature.

RWIS Weather Features (RW): Unlike traffic accidents, weather features such as temperature are continuously distributed over the entire space. Therefore we find the k-nearest RWIS stations to each grid cell s_i in **Euclidean distance**. Then, we calculate the average measure of the three stations in each hour as the hourly estimated measure of s_i . Finally we extract the average hourly measures at s_i for day t as feature values. Specifically, the feature x (e.g., temperature) at grid cell s_i for day t is

$$RW_x(s_i, t) = \frac{1}{24} \Sigma_{h=0}^{23} \Big[\frac{1}{|N_{s_i}|} \Sigma_{q \in N_{s_i}} RW_x(q, h) \Big] \tag{1}$$

where q is a RWIS station, h is an hour of day t, N_{s_i} is the set of k nearest q of s_i. In our implementation we use $k = 3$. This group includes 4 features: temperature, wind speed, dew point.

Traffic Volume Features (V): In addition, we obtain real-time traffic volume information from the Iowa DOT traffic cameras in the entire state of Iowa. For each grid cell, we identify the three nearest cameras using **network distance**, on the road network. Each camera records the total number of vehicles passed during each hour along both directions.

Calendar Features (CL): We finally add calendar features for each day, including the day of year (1 to 365), day of week (1 to 7), month of year (1 to 12), quarter of year (1 to 4) and If_Holiday (0 or 1). These features are non-spatial features, i.e., same value for all the locations for each day.

4.3 SpatialGraph Features

To tackle the spatial heterogeneity, we also take the spatial relationship of the roads into account. Although the spatial heterogeneity can be captured to some degree by road specific and weather related features, there are still many factors that could make the accidents occurring pattern different in different areas. For example, from Fig. 1 (a) we can see that more accidents are concentrated in urban areas (e.g., Des Moines, Cedar Rapids) than in rural areas, which can be attributed to different population density in different areas.

One solution is to include a new set of features that consider the spatial relationship between different locations. The idea is to construct a spatial graph between all roads and to conduct the eigen-analysis of the induced Laplacian matrix [6]. We obtain the resulting top eigen-vectors of the Laplacian matrix. These eigen-vectors provide additional information about the topological feature of each road with respective to potential spatial clusters in the road network. Specifically, let $L \in \mathbb{R}^{m \times m}$ denote the graph Laplacian matrix computed based on the spatial graph, where each row of L corresponds to a road segment in the data. Let $V \in \mathbb{R}^{m \times K}$ denote the top K eigen-vectors of L. Then we can use each row of V to induce a new set of features for the corresponding road segment. This approach is similar to spectral clustering [30], which first generates the eigen-features based on the Laplacian matrix and then conduct the k-means clustering based on the new features. We also use the spectral clustering to visualize the generated features based on the clustering results in Figure 1 (k=10). The feature construction process is summarized in Algorithm 1. Note the last step is different from spectral clustering.

The above process will generate k spatialGraph features for each road segment. Finally we map the features to each grid cell s_i. For grid cells with only one road segment, the spatialGraph features are directly assigned as those of the corresponding road segment. For grid cells with multiple road segments, we pick the longest road segment in s_i and use its spatialGraph features for s_i. This approximation is reasonable because road segments in the same grid cell are typically connected and tend to have very similar spatialGraph features. In our implementation we choose $k = 10$, resulting in 10 time-invariant spatialGraph features.

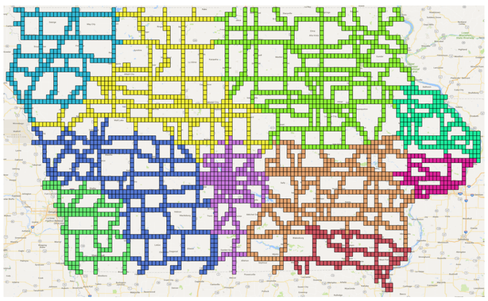


Figure 2: Visualization of Spectral Clustering when K = 10 (best viewed in color).

Algorithm 1: SpatialGraph Feature Construction

Data: Road Network $N = \{Rd_1, Rd_2, ..., Rd_m\}$, spatial grid S, the number of features desired k

Result: k SpatialGraph features $E \in \mathbb{R}^{|S| \times k}$ for all grid cells in S

Construct an adjacency matrix W of N and a degree matrix D of N;

Compute the normalized Laplacian $L_{norm} = D^{\frac{-1}{2}} W D^{\frac{-1}{2}}$;

Compute the first k eigenvectors $v_1, ..., v_k$ of L;

Let $V \in \mathbb{R}^{m \times k}$ be the matrix containing the vector $v_1, ..., v_k$ as columns;

For $i = 1, ... ,m$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i_{th} row of V;

For $s_j \in S$, $E(s_j) \leftarrow y_i$, where Rd_i is the longest road $s.t. Rd_i \cap s_j \neq \emptyset$

4.4 Summary of Features

We finally constructed 31 features, which are grouped into 7 categories: Road Network (N) (1 feature), Road Condition (RC) (6 features), Satellite Image (G) (3 features), Rainfall (RA) (1 feature), Weather (RW) (4 features), Traffic Volume (V) (1 features), Calendar Features (CL) (5 features), and SpatialGraph (E) (10 features). Each feature is converted to a three-dimensional 64×128×1 tensor.

5 THE HETERO-CONVLSTM APPROACH

This section presents our solutions to the traffic accident prediction problem. We first introduce the Convolutional LSTM and then discuss how we build the Hetero-ConvLSTM model with ConvLSTMs.

5.1 Convolutional LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network node structure known to have good performance when handling time series data with temporal auto-correlations. A node in a LSTM Neural Network consists of a memory cell, an input gate, an output gate, and a forget gate. During the training phase, a weighted function is learned in each of the gates in a LSTM node to control the “memorizing” and “forgetting” capability of the network.

The ConvLSTM model is a variation of LSTM to handle spatio-temporal prediction problems, which was first introduced by Shi et al. [32] for precipitation nowcasting. Each input feature of a ConvLSTM network is a three-dimensional spatio-temporal tensor, where the first two dimensions are the spatial dimension. Comparing with the original LSTM model, the input-to-state and state-to-state transitions of the ConvLSTM cell involves convolutional operations that outputs 3-dimensional tensors. This model can be further formulated as the following equations. The $*$ denotes the convolution operation and $\circ$ denotes the Hadamard product.

$$i_t = \sigma\big(W_{xi} * X_t + W_{hi} * h_{t-1} + b_i\big) \tag{2}$$

$$f_t = \sigma\big(W_{xf} * X_t + W_{hf} * h_{t-1} + b_f\big) \tag{3}$$

$$o_t = \sigma\big(W_{xo} * X_t + W_{ho} * h_{t-1} + b_o\big) \tag{4}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ tanh\big(W_{xc} * X_t + W_{hc} * h_{t-1} + b_c\big) \tag{5}$$

$$h_t = o_t \circ tanh\big(C_t\big) \tag{6}$$

In the equations, i_t, f_t, o_t are the outputs of input gate, forget gate, and output gate for time step t. C_t is the cell output at time step t. h_t is the hidden state of a cell at time step t.

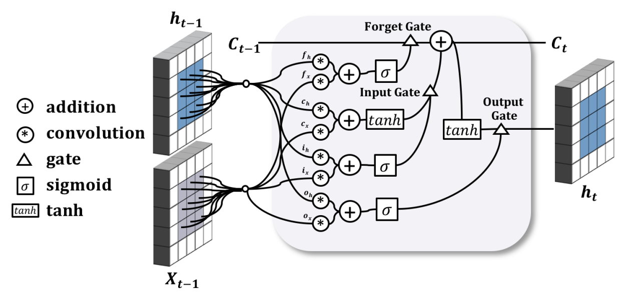


Figure 3: The inner structure of a ConvLSTM cell.

The CovnLSTM has nice properties for traffic accident prediction as the LSTM part may capture temporal auto-correlation in the data and the convolution operator may capture local spatial features (e.g., dangerous road intersections) that are important indicators of potential accidents. However, ConvLSTM does not handle spatial heterogeneity explicitly. Although we incorporate spatialGraph features as detailed in Section 4.3, model accuracy might be affected due to varying environmental conditions. Also training a single, large ConvLSTM model may require excessive time.

To address the above limitations, we propose the Hetero-ConvLSTM framework. First, we use a moving window to obtain the data of a sub-region in the study area and learn a ConvLSTM network model for each window with different parameters. The size of the window is chosen such that the model can be trained in reasonable time, while the region is still large enough to include sufficient training samples. In our case, we choose a region with size 32×32.

After the models are trained, we obtain the prediction for every grid from every model and use the ensemble of the outputs as the final prediction.

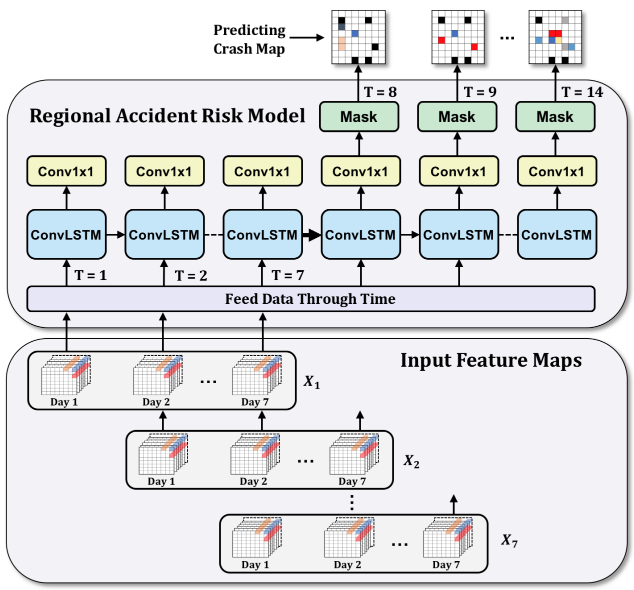


Figure 4: The structure of the regional ConvLSTM model.

5.2 Regional Prediction Model

ConvLSTM Structure: For each regional window, we construct a ConvLSTM model. In spatio-temporal prediction such as rainfall forecasting [32], there is only a single dependent variable but no other features. The inputs X are the historical values of the dependent variable. In our problem, we have additional features incorporated and fed to the network. Therefore, in our problem, the input data is 4-dimensional. The parameters X_t and h_{t-1} in the above formulas are three-dimensional tensors rather than two-dimensional images as in [32].

Four ConvLSTM layers are stacked, where each layer has 128 ConvLSTM filters (hidden states) to extract spatial features from input data and the output of the previous time steps. Between any two ConvLSTM layers, we apply batch normalization layer to further accelerate the training processes. For pixel-wise prediction, we concatenate all outputs and feed them into a 1 x 1 convolutional forward layer to generate a two-dimensional map for each time step t.

We also implement an additional filter on the final output to smooth the outputs. The final output image for each time step t is filtered by the network mask layer (N) through a pixel-wise AND operation. Any non-zero predicted value outside of the road network are set to 0.

Figure 4 shows the overall architecture with the inputs and outputs. Fig 3 shows the detailed structure of a single ConvLSTM cell in our model.

Training and Testing: We construct the training samples in the following way. Each training sequence consists of 14 days, where the last 7 days are predicted based on the data in the first 7 days (last 7 days are shifted by 1 day from first 7 days). All the feature tensors of the 14 days are fed into the ConvLSTM model. We choose 7 days because traffic accidents are impacted by human activities, which have a strong weekly pattern.

The model generates the prediction for one day t at a time based on the features (including ground truth accident counts) up to day $t-1$. Then the ground truth $C(S,t)$ is given as the input to the next prediction $C(S,t+1)$. Note we do not use any feature on day t because these features may not be available before day t has passed. All the features described in Section 4 are normalized to the range of [0,1] before being used. We use cross-entropy as the loss function:

$$Loss = -\Sigma_{s,t} T_{s,t} log P_{s,t} + (1 - T_{s,t}) log(1 - P_{s,t}) \tag{7}$$

where $T_{s,t}$ and $P_{s,t}$ are ground-truth and predicted crash map of location s at time t.

5.3 Spatial Ensemble of ConvLSTM Models

In order to address the spatial heterogeneity problem, we build an LSTM model for each different regions in the study area and use a model ensemble method to generate the final results. The intuition behind this design is to reduce the impact of data heterogeneity (e.g., urban vs. rural). We use a moving window approach, where the size of the moving window is 32×32. We take subsets of the spatial framework S by moving the window from the top-left corner (0,0)-(32,32) to the bottom-right corner (96,32)-(128, 64), with a step of 16 grids on both of the horizontal and vertical directions. This results in 21 different regions, where every pair of neighboring regions have 50% grids overlapping with each other. The regions are illustrated in Figure 5.

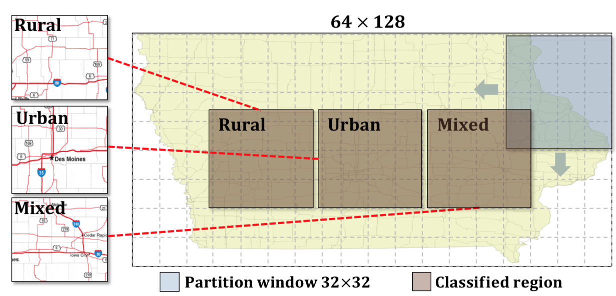


Figure 5: Map partitioning of spatial ensemble model (stride=16).

We split the training set and the testing set each into 21 subsets accordingly. For each regional window, we learn a ConvLSTM model based on its own training set, and make predictions on its testing dataset. The final prediction of a grid location s_i at each day t_j is calculated as a weighted average of predicted values at s_i on t_j from all the models, whose regions cover s_i. Formally,

$$\hat{C}(s_i, t_j) = \frac{1}{\Sigma w} \Sigma_{k=1}^{N} \left(w_k \hat{C_{W_k}}(s_i, t_j) \times I(s_i \in W_k)\right) \tag{8}$$

where N is the total number of windows that cover s_i, and w_k is the weight for window W_k. The optimal weights could be learned through a linear regression of regional model outputs. However for simplicity in this paper we choose equal weights for each w.

6 EXPERIMENTAL RESULTS

6.1 Experiment Settings

Data Preparation: We formulate our problem as prediction on next 7-day traffic accident based on traffic accident and other related conditions over last 7 days. Thus, we create each training and testing sample as a sequence of 14 frames (7 frame for the training and next 7 frames for the prediction). The entire dataset (8 years, 2922 days) is converted to 2915 sequences. All data are divided into 2 groups, where the data for the first 7 years (2006-2012) is used as training set, and the data in the last year (2013) is used as the testing set. 10% of the training set is selected to be the validation set.

As described in the previous section, we partition the state of Iowa into 5km by 5km grids. For each day in 2013 we predict a traffic accident map using the proposed Hetero-ConvLSTM model.

Evaluation Tasks: Through the experiments we wish to answer the following questions: (1) Are the results of the proposed framework better compared with baseline methods, including classical predictive models and ordinary ConvLSTM? (2) Which features have the most impact on the prediction accuracy in different regions? (3) How does the performance of our proposed model vary on different regions (e.g., urban, rural)? (4) Do the prediction results make sense? Are the predicted accident locations correlated with the ground truth spatially?

Evaluation Metrics: We evaluate the accuracy of the models by using the following measures: mean squared error (MSE), root-mean-square error (RMSE), Cross Entropy (CE). We also use the Cross-K function [13] to evaluate the spatial correlation between the predicted results and the ground truth.

Parameter Configurations: We train the proposed Hetero-ConvLSTM models by minimizing cross-entropy loss using Adam optimizer with the settings: $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. We also employ early-stopping technique during training. We also identify a cut-off threshold θ, where predicted values below θ are set to zero. This helps remove very small predicted values, which are very unlikely to be real events and are negligible. We use $\theta = 0.0595$ through tuning on the validation set.

Platform: We set up the experiments on Argon High Performance Computing System at the University of Iowa [20] using a 256GB RAM computing node with 2.6GHz 16-Core CPU. For the training of deep learning models, we use GPU node on Argon with Nvidia Tesla P100 Accelerator Cards with the support of Tensorflow library.

Baseline Models: We compare our proposed framework with the following baseline models: (1) Least Squares Linear Regression (**LR**), (2) Decision Tree Regression (**DTR**), where the Information Gain in ID3 algorithm by Ross Quinlan [25] is replaced by Standard Deviation Reduction. (3)**DNN**. A two-layer fully-connected neural network of 2048 hidden units each. In addition, we apply dropout regularization to prevent overfitting. (4) **FC-LSTM**. For the FC-LSTM, we use a two-layer LSTM network structure consisting of 2048 memory cells per layer. (5) **ConvLSTM**. Ordinary ConvLSTM without spatial ensemble. (6) **Historical Average**. Average daily accident counts over 7 years.

6.2 Results on Prediction Accuracy

Considering spatial heterogeneity, we present the results in three different types of regions in the experiments. We define **Type-1** as urban region, e.g., Des Monies, **Type-2** as rural region, e.g., western Iowa, and **Type-3** as mixed region (between two urban regions), e.g. region among, Waterloo, Iowa City and Cedar Rapids. We report the evaluation measures of results on each region (in pixel level).

The results are presented in table 1. The Hetero-ConvLSTM model achieves the best performance with the lowest MSE and RMSE in all the three regions. Historical Average performs better than other baseline models but worse than our proposed model in all the regions. It reveals that daily average is generally a good estimator for long-term prediction with low average error due to the periodicity and seasonality pattern of traffic accidents. However it might not be good at predicting short-term traffic accidents, especially when accidents are caused by weather condition or rare events. In summary, the Hetero-ConvLSTM is an order of magnitude more accurate than all the other models.

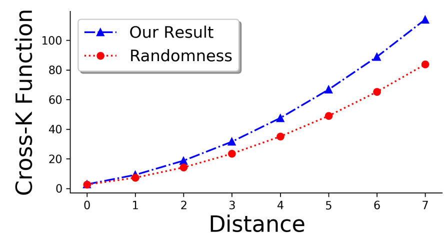


Figure 6: Cross-K function between predicted and actual accidents.

6.3 Impact of Feature Groups

To investigate the effect of different features group on the results, we run our trained model on all the 32x32 regions and present the results on the three selected regions as well as the overall results of the entire state. For each region, we add 1-2 feature groups at a time and measure the MSE, RMSE, and Cross-Entropy of the results. The results are summarized in table 2. We can observe that generally more features lead to lower errors. However, some feature groups might affect the accuracy negatively and the most important features differ in different regions.

In Type-1 (urban), Road Network (N), Traffic Volume (V), Road Condition (RC), and Calendar (CL) bring down errors. This can be explained by higher density of population and stronger human activity patterns. SpatialGraph features (E) have a very weak impact. This might be due to lower heterogeneity of the area.

Interestingly, in Type-2 and Type-3 regions, the important feature groups changed. Rainfall (RA), other weather features such as temperature and wind speed (RW), and SpatialGraph features (E) bring down the error more effectively than other features. This makes good sense since human related factors are less important than environmental features in rural areas. The results also verify that strong spatial heterogeneity exist in the data (especially rural regions) and our proposed SpatialGraph features contributed to handling this challenge. The overall results show that over the entire state, calendar (CL) and Road Condition (RC) features have the most contributions on the accuracy of the results. However, the impacts are marginal, due to spatial heterogeneity.

Table 1: Model Performance

Model	Type-1 Urban			Type-2 Rural			Type-3 Mixed		
	MSE	RMSE	CE	MSE	RMSE	CE	MSE	RMSE	CE
LR(C=0.1)	0.146	0.382	0.051	0.040	0.199	0.002	0.086	0.294	0.014
DTR(depth=30)	0.172	0.415	0.243	0.056	0.237	0.123	0.111	0.334	0.230
DNN(2048x2048)	0.140	0.374	0.033	0.036	0.190	0.023	0.082	0.286	0.011
FC-LSTM(2048x2048)	0.187	0.434	0.419	0.042	0.205	0.419	0.089	0.298	0.001
ConvLSTM (128x128x128x128)	0.117	0.343	0.074	0.037	0.192	0.025	0.077	0.278	0.071
Historical Average (7 years)	0.050	0.224	0.340	0.015	0.121	0.219	0.033	0.181	0.295
Hetero-ConvLSTM (128x128x128x128)	**0.021**	**0.144**	**0.014**	**0.006**	**0.078**	**0.001**	**0.013**	**0.116**	**0.010**

Table 2: Impact of Feature Groups

Model	Type-1 Urban			Type-2 Rural			Type-3 Mixed			All Regions		
	MSE	RMSE	CE	MSE	RMSE	CE	MSE	RMSE	CE	MSE	RMSE	CE
N	0.120	0.346	0.089	0.063	0.251	0.212	0.082	0.286	0.068	0.049	0.222	0.047
N+RW+RA	0.126	0.356	0.073	0.038	0.195	0.046	0.076	0.276	0.087	0.056	0.237	0.074
N+RW+RA+V+RC	0.123	0.351	0.127	0.039	0.199	0.006	0.100	0.316	0.256	0.049	0.221	0.037
N+RW+RA+V+RC+G	0.148	0.384	0.247	0.038	0.194	0.039	0.080	0.283	0.050	0.048	0.219	0.043
N+RW+RA+V+RC+G+CL	0.118	0.344	0.075	0.046	0.216	0.100	0.082	0.286	0.018	0.048	0.220	0.030
N+RW+RA+V+RC+G+CL+E	0.117	0.343	0.074	0.037	0.192	0.025	0.077	0.278	0.071	0.049	0.222	0.026

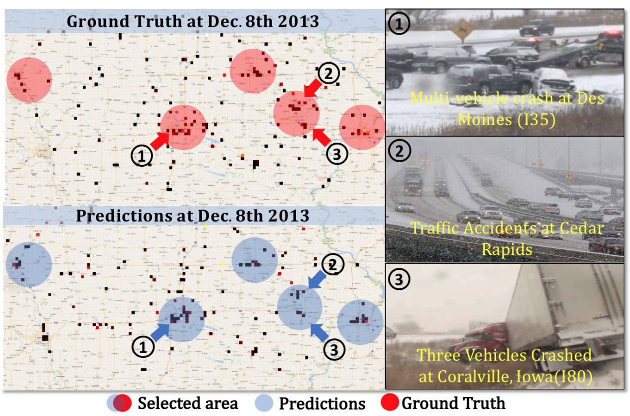


Figure 7: Case study of traffic accidents on Dec. 8th, 2013. From top to bottom: ground truth and predictions using Hetero-ConvLSTM.

6.4 Result Quality: A Case Study

Finally, we evaluate the location accuracy of the prediction results using the cross-K function [13]. Cross-K function is a measure of spatial clustering tendency between two object types. In our case, we calculate on average the density of predicted accidents within every distance d of a real accident. Varying d we obtain an empirical curve. Figure 6 shows the curves. The bottom curve represents the cross-K function for complete spatial randomness, where there is no correlation between the ground truth and the prediction. In this case $K(d) = A_p(2d+1)^2/A_g$, where A_p and A_g are spatial density of predicted and real accidents. The top curve represents the empirical cross-K function between Hetero-ConvLSTM predictions and the ground truth. Results show that the prediction is highly correlated with the ground truth spatially.

We visually compare the results of Hetero-ConvLSTM with the ground truth. Figure 8 shows the prediction result of a whole week over one of the selected regions (Type-3). The circles highlight regions where the patterns are correctly predicted. As can be observed, our model is able to find the major hotspots of accidents. The trends in the predictions match with those in the ground truth data.

Finally, we identify from news report that on December 8, 2013, a big snow storm attacked Iowa. More than 49 accidents were reported in Cedar Rapids area, close to the southbound lane of I-380 [1]. Figure 7 shows the predicted results vs. the ground truth on this day. As can be observed, clusters of accidents including the aforementioned one are correctly predicted.

7 SYSTEM DEPLOYMENT

Our framework presents a solution to predicting daily/weekly traffic accident risk using big heterogeneous data. Our evaluation results are carried out on an experimental prototype system. To fully deploy a working system accessible to the public we need to address the following challenges: 1) deploy our framework on a cloud-based service/storage for real-time query. 2) update the offline model periodically to keep the predictions accurate.

The feasible solution to first problem is to deploy our trained model in a cloud service (e.g., Amazon AWS Lambda [2] or Microsoft Azure [4]). For daily prediction, our system will be scheduled to run and fetch the weather, traffic volume, rainfall, and crash data of the past day at midnight, transform them into feature maps and then retrain the model in an online manner. The learned model parameters are finally pushed and saved to the cloud service. For weekly prediction, the only difference is to use simulated or forecast data for time-variant features. Ideally, we will provide a web interface for user to view the crash risk map in one or more days. In future, we also consider the possibility of incorporating this framework with current map service platform, such as Google Map [3].

8 CONCLUSION

This paper investigated the problem of traffic accident forecasting using deep learning models on heterogeneous urban data. This is an important problem to transportation and public safety. It is

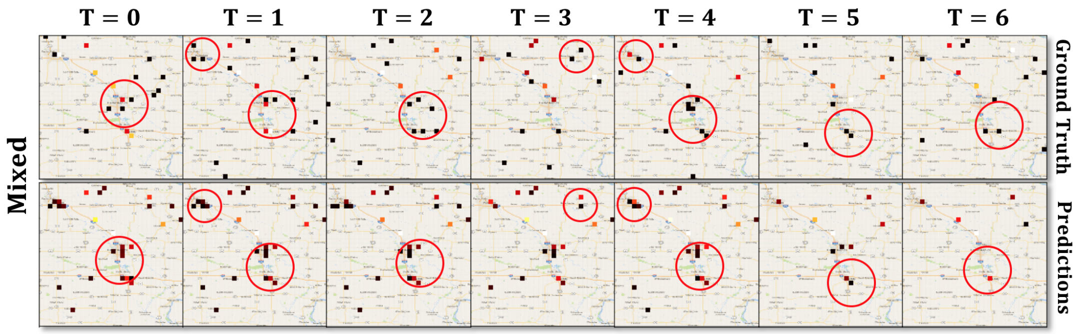


Figure 8: Qualitative results on traffic crash prediction on Type-3 region. Red circle shows selected areas for comparison. Different color represents different level of traffic crash. Darker color means higher crash counts (best viewed in color).

also challenging due to the trade-off between sparsity and spatial heterogeneity of data. In this paper we performed a detailed study on the traffic accident prediction problem using the Convolutional Long Short-Term Memory (ConvLSTM) neural network. A number of urban and environmental features were extracted from big datasets over the state of Iowa across 8 years. We proposed a Hetero-ConvLSTM framework with spatial graph features and spatial model ensemble to address the spatial heterogeneity challenge. Experiments on the 8-year data over the entire state of Iowa showed that the proposed Hetero-ConvLSTM outperforms all the baselines in prediction accuracy.

This work showed that deep learning techniques such as ConvLSTM are promising solutions to traffic accident prediction if unique data properties such as spatial heterogeneity are well handled.

ACKNOWLEDGMENTS

The work of this paper is partially supported by National Science Foundation under grant number IIS-1566386. We thank the Injury Prevention Research Center (IPRC) at the University of Iowa for providing part of the motor vehicle crash data in the early stage of this work. We also thank Data Mining at Iowa Group (DMIG) for their valuable comments.

REFERENCES

[1] 2013. Snow blankets Eastern Iowa causing slick conditions, http://www.thegazette.com/2013/12/08/snow-blankets-eastern-iowa-causing-slick-conditions. *The Gazette* (2013).
[2] 2018. Amazon AWS Lambda. (2018). https://aws.amazon.com/lambda/
[3] 2018. Google Map. (2018). https://www.google.com/maps
[4] 2018. Microsoft Azure. (2018). https://azure.microsoft.com/en-us/
[5] 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.
[6] Mikhail Belkin and Partha Niyogi. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, Vol. 14. 585–591.
[7] Ruth Bergel-Hayat, Mohammed Debbarh, Constantinos Antoniou, and George Yannis. 2013. Explaining the road accident risk: Weather effects. *Accident Analysis & Prevention* 60 (2013), 456–465.
[8] Ciro Caliendo, Maurizio Guida, and Alessandra Parisi. 2007. A crash-prediction model for multilane roads. *Accident Analysis & Prevention* 39, 4 (2007), 657–670.
[9] Li-Yen Chang. 2005. Analysis of freeway accident frequencies: negative binomial regression versus artificial neural network. *Safety science* 43, 8 (2005), 541–557.
[10] 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.
[11] Quanjun Chen, Xuan Song, Harutoshi Yamada, and Ryosuke Shibasaki. 2016. Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident Inference.. In *AAAI*. 338–344.
[12] Miao M Chong, Ajith Abraham, and Marcin Paprzycki. 2004. Traffic accident analysis using decision trees and neural networks. *arXiv preprint cs/0405050* (2004).
[13] Noel Cressie. 2015. *Statistics for spatial data*. John Wiley & Sons.
[14] Iowa DOT. 2016. GIS REST Services: Crash Data. (2016). https://gis.iowadot.gov/public/rest/services/Traffic_Safety
[15] Daniel Eisenberg. 2004. The mixed effects of precipitation on traffic crashes. *Accident analysis & prevention* 36, 4 (2004), 637–647.
[16] Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. 2017. Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment* (2017). https://doi.org/10.1016/j.rse.2017.06.031
[17] 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.
[18] Ying Lin and Kenneth E Mitchell. 2005. 1.2 the NCEP stage II/IV hourly precipitation analyses: Development and applications. In *19th Conf. Hydrology*. Citeseer.
[19] Alameen Najjar, Shun'ichi Kaneko, and Yoshikazu Miyanaga. 2017. Combining Satellite Imagery and Open Data to Map Road Safety.. In *AAAI*. 4524–4530.
[20] University of Iowa. 2016. Neon Overview and Quick Start Guide. (2016). https://wiki.uiowa.edu/display/hpcdocs/Neon+Overview+and+Quick+Start+Guide
[21] Department of Transportation. 2016. Iowa DOT GIS REST Services: Road Network. (2016). https://gis.iowadot.gov/public/rest/services/RAMS
[22] Department of Transportation. 2016. The Iowa Environmental Mesonet: Roadway Weather Information System. (2016). https://mesonet.agron.iastate.edu/RWIS/
[23] Jutaek Oh, Simon P Washington, and Doohee Nam. 2006. Accident prediction model for railway-highway interfaces. *Accident Analysis & Prevention* 38, 2 (2006), 346–356.
[24] Dev Oliver, Shashi Shekhar, Xun Zhou, Emre Eftelioglu, Michael R Evans, Qiaodi Zhuang, James M Kang, Renee Laubscher, and Christopher Farah. 2014. Significant route discovery: A summary of results. In *International Conference on Geographic Information Science*. Springer, 284–300.
[25] J. Ross Quinlan. 1986. Induction of decision trees. *Machine learning* 1, 1 (1986), 81–106.
[26] Honglei Ren, You Song, JingXin Liu, Yucheng Hu, and Jinzhi Lei. 2017. A Deep Learning Approach to the Prediction of Short-term Traffic Accident Risk. *arXiv preprint arXiv:1710.09543* (2017).
[27] Lei Shi and Vandana P Janeja. 2011. Anomalous window discovery for linear intersecting paths. *IEEE Transactions on Knowledge and Data Engineering* 23, 12 (2011), 1857–1871.
[28] 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.
[29] World Health Organization. Violence, Injury Prevention, and World Health Organization. 2015. *Global status report on road safety 2015: supporting a decade of action*. World Health Organization.
[30] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17, 4 (2007), 395–416.
[31] Jing Wang and Xin Wang. 2011. An ontology-based traffic accident risk mapping framework. In *International Symposium on Spatial and Temporal Databases*. Springer, 21–38.
[32] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*. 802–810.