Predicting Traffic Accidents Through Heterogeneous Urban Data: A Case Study

Zhuoning Yuan
University of Iowa
zhuoning-yuan@uiowa.edu

Xun Zhou
University of Iowa
xun-zhou@uiowa.edu

Tianbao Yang
University of Iowa
tianbao-yang@uiowa.edu

James Tamerius
University of Iowa
james-tamerius@uiowa.edu

Ricardo Mantilla
University of Iowa
ricardo-mantilla@uiowa.edu

ABSTRACT

With the urbanization process around the globe, traffic accidents have undergone a rapid growth in recent decades, causing significant life and property losses. Predicting traffic accidents is a crucial problem to improving transportation and public safety as well as safe routing. However, the problem is also challenging due to the imbalanced classes, spatial heterogeneity, and the non-linear relationship between dependent and independent variables. 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. This paper, through a case study, presents our explorations on effective techniques to address the above challenges for better prediction results. Specifically, we formulate the problem as a binary classification problem. For each road segment in each hour, we predict whether an accident will occur. Big data including all the motor vehicle crashes from 2006 to 2013 in the state of Iowa, detailed road network, and various weather attributes at 1-hour granularity have been collected and map-matched. We evaluate four classification models, i.e., Support Vector Machine (SVM), Decision Tree, Random Forest, and Deep Neural Network (DNN). To tackle the imbalanced class problem, we perform an informative negative sampling approach. To tackle the spatial heterogeneity challenge, we incorporate SpatialGraph features through Eigen-analysis of the road network. Results show that employing informative sampling and incorporating the SpatialGraph features could effectively improve the performance of all the models. Random Forest and DNN generally perform better than other models.

CCS CONCEPTS

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

UrbComp-2017, August 2017, Halifax, Nova Scotia, Canada

ACM ISBN 123-4567-24-567/08/06...$15.00
https://doi.org/10.475/123_4

KEYWORDS

traffic accident prediction; big data; eigen-analysis; spatial heterogeneity

ACM Reference format:
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 6th International Workshop on Urban Computing, Halifax, Nova Scotia, Canada, August 2017 (UrbComp-2017),* 9 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

Traffic accidents have been a significant issue to public safety. The total traffic crash deaths reached 1.25 million in 2013 globally [25]. The ability to predict future accidents (e.g., where, when, or how) is thus very useful not only to public safety stakeholders (e.g., police) but also transportation administrators and individual travelers. A potential application of such technique would be real-time safe route recommendation for drivers. With the rapid development of data collection techniques and the availability of big urban datasets in recent years, predicting traffic accidents has become more realistic. Detailed rainfall data, public transportation information, and motor vehicle crash reports could provide valuable information for traffic accident analysis.

However, this problem is very challenging due to a few issues. (1) Class imbalance. Traffic accidents are rare incidents. If we construct class labels based on accident vs. no-accident for each road, the classes will be severely imbalanced. (2) Spatial heterogeneity, i.e., the prediction model parameters may vary from place to place. For example, factors causing traffic accidents in large cities with dense population and lower speed limits might be very different from those in rural areas with low population density but high speed limit. A global model might not be very accurate everywhere. (3) The relationship between environmental factors and accidents might be complex and non-linear. Simple linear models might not achieve good performance.

In recent years, many machine learning methods have been successfully applied on urban computing problems such as traffic volume prediction [10, 28]. However, traffic accident prediction has received attention mostly from domain researchers. Previous research on traffic accident prediction typically use small datasets with limited features. Also they

usually directly apply classical prediction models on the data, without properly addressing issues such as spatial heterogeneity and class imbalance. Finally, most of these work lack a comprehensive evaluation of their results (mostly Accuracy).

This paper presents our explorations on effective ways to improve traffic accident prediction results, which is an essential step towards building robust and reliable traffic accident predictive models. Specifically, we consider a binary classification problem. For each road segment at each hourly time slot, we aim to predict whether there will be any accident or not. We collect fine-grained big datasets, including all the motor vehicle crashes in the state of Iowa between 2006 and 2013, detailed road network, and hourly weather data such as rainfall, air temperature, etc. To address the limitations of related work, we use informative sampling strategies to mitigate the class imbalance problem. We also incorporate spatial relationship of the roads into the predictive models through an eigen-analysis on the road network to improve the prediction performance. We conduct experiments on four classical classification models, i.e., Support Vector Machine, Decision Tree, Random Forest, and Deep Neural Network.

We highlight our contributions as follows:

1. We collect and fuse heterogeneous urban 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.

2. 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 also propose an informative sampling approach to construct negative samples to mitigate class imbalance problem.

3. We perform comprehensive experiments on four classification models, i.e., SVM, Decision Tree, Random Forest, and Deep Neural Network (DNN). Results show that our proposed approaches effectively improve all the measures for all the models. The best Accuracy and AUC we achieved are 0.9512 and 0.9612, respectively.

The rest of the paper is organized as follows: Section 2 discusses the related work of this paper. Section 3 presents data collection and pre-processing steps. Section 4 proposes our approach on feature engineering, sampling, and the classification models. Section 5 presents experiment results. Finally Section 6 concludes the entire paper and discusses future work.

2 RELATED WORK

We classify the related work into two types, as discussed below.

Classification: The most relevant work to ours are those using classification models. They aim to classify each given road segment at given time into binary classes {Accident, No Accident}. Chang [5] 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. [6] 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. [8] applied decision tree and ANN model on a dataset from Nigeria and achieved precision and recall both around 0.52. Lin et al. [13] 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%. AbellÃąn et al. [1] used a Probabilistic Neural Network (PNN) model to predict traffic accident based on real-time road condition (e.g., traffic volume, speed). The best accuracy achieved is around 70%. Although these work used different data sets therefore directly comparing accuracy is unfair, there is still much we could do to improve the performance. First, the above work are simple application of classical prediction models by domain researchers. Most of them only used one model. Also they have not addressed key issues such as the class imbalance problem and spatial heterogeneity. Our work, by contrast, incorporates spatial structure of the road network into the classifier through eigen-analysis and significantly improves the performance of all the models examined.

Numeric Prediction and Correlation Analysis: The second group of works aim at fitting regression or other models to predict the number of traffic accident 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. Chen et al. [7] developed a deep learning model to predict the traffic accident risk level using human mobility data. Caliendo et al. [4] developed Poisson, Negative Binomial, and Negative Multinomial regression models to predict the number of accidents on given roads. Oh et al. [20] 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. [3] employed an auto-regressive regression model to study the correlations between weather attributes and injury accidents. Eisenberg et al. [9] used negative binomial regression model to study the relationship between monthly precipitation and monthly fatal crashes. Tamerius et al. [23] analyzed the Relative Accident Rate (RAR) to study the relationship between precipitation and motor vehicle crashes over space and time. All these work, however, are very different from our work as their outputs are numeric values rather than binary class labels. Their results are not directly comparable with ours.

3 DATA SOURCES AND PRE-PROCESSING

3.1 Data Sources

Motor Vehicle Crash Data. We obtained motor vehicle crash data from the Iowa Department of Transportation (DOT) [17]. 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. For simplicity we round the crash time to the nearest hour before the crash, e.g. 12:36 becomes 12:00. Figure 4(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 [14]. 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 4(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) [19] . 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 4(c) shows the locations of the observation stations.

Road Networks. We collected three different road network datasets from Iowa DOT GIS data portal [18] with basic road information in the state of Iowa, detailed speed limits of the road, and the the most recent estimated Annual Average Daily Traffic (AADT) volume for the primary roads. The AADT data also include detailed statistics of each type of vehicles, such as Single Unit Truck AADT and Combination Truck AADT, for the secondary roads.

Demographic Data. We retrieved census data of Iowa from Geographic Information Systems Library. The Census data divided State of Iowa into 826 small census tracts. Each of them represents the sub-area of a county or a smaller area defined by local participants, which usually has a population between 1,200 and 8,000. We only use the population and area size information of each census block.

3.2 Interpolation of Missing Values

The weather related features collected by the RWIS project contain many missing values (e.g., the air temperature, dew point temperature, wind speed, gust speed). Instead of removing all records that have missing values, we first try to impute the missing values by interpolation. The underlying assumption is that the weather related features are usually auto-correlated and thus change smoothly over space. Therefore, if a station contains a missing value for a particular feature at a certain time point, we interpolate it using the observations of the same feature at the same time point from nearby stations by inverse distance weighting (IDW) [27]. The IDW estimation of a missing value is computed by

$$Z = \frac{W_i Z_i}{W_i}$$

where $W_i = \frac{1}{d_i^2}$, d_i denotes the distance from a nearby station to the target station, and Z_i is the value of the feature at the nearby stations. We used data from three nearest stations to estimate the missing value of a station. If none of them is available, we deem this record as an example with missing values. Finally, all examples with missing values after imputation are removed. We also verify the accuracy of imputation by a ***leave-one-out*** approach. We use the data of three nearby stations to estimate each available feature value and then calculate the average difference between the estimated values and the observed values. The results show the average errors of eight years RWIS data for *air temperature*, *dew point temperature*, *gust speed*, *wind speed* are $1.99^\circ F$, $2.69^\circ F$, 1.97 (mph), 2.19 (mph), respectively, which are relatively small compared to the average values of these features in the data and thus verify that our imputing method is accurate.

3.3 Map Matching

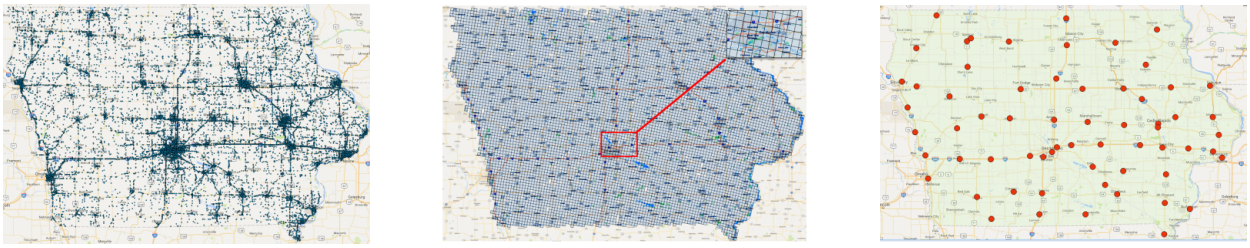
In order to generate the input data for our prediction, we need to match each crash with the corresponding road, traffic, and weather information at the crash location and time. This task is challenging due to various data types and large data volume.

Combining Network Datasets: One of the most challenging task of our work is to merge three different road networks into one. Due to the different mapping criteria and time, the geometry shapes of the the same road in the three datasets do not completely match. We study the road relationships among different networks and found three types of topological relationships between the representations of the same road in two different road networks:

- Relationship 1: A road in Network A is a bit longer than a similar road in Network B.
- Relationship 2: A road in Network A is comprised of several shorter roads in Network B, or vice versa.
- Relationship 3: A road in Network A partly intersects with a road in network B.

We implement a series of spatial joins in PostgreSQL to do the network data matching. The steps are as follows: for two road network datasets A and B, We first generate a buffer for each road in dataset A. If a road segment b in dataset B is contained in the buffer of a road a in A then we consider a and b are the same road and merge b's features into a's record. We swap A and B and repeat the above step. All the matched roads are removed from the datasets to avoid duplicate matches. Finally we find pairs of road segments in A and B, whose buffers intersect with each other. and merge each pair into a new road segment.

Matching Crash Data with Road Network: We match each crash event with the nearest road segment generated in the previous step, if the distance between them is within a minimum distance and road network features are available. Otherwise, the crash is considered an outlier and removed. We test different minimum distance and found 35m to be the suitable value. Totally we matched 96% of all the accidents in the data.



(a) Visualization of Traffic Accidents of 2013 (b) Rainfall Map (c) RWIS Observation Stations

Figure 1: Illustrations of the Motor Vehicle Crash, Rainfall, and RWIS datasets

Matching Weather Data with Road Network: In the Radar Rainfall data, the entire state of Iowa is partitioned into square tiles of 4km by 4km. For each road segment, we find the tiles that overlap with it and use the average rainfall amount of these tiles as the actual rainfall for this road.
The RWIS dataset is only available from 63 stations along major highways in Iowa. We construct a Voronoi Diagram to partition the state of Iowa. Figure 5 shows the Voronoi Diagram where red dots are the location of the weather stations. For each road segment, we find the Voronoi partitions that overlap with it, and use the average value from the corresponding stations as the weather data for this road segment. If none of these stations have available data we mark the value as missing.

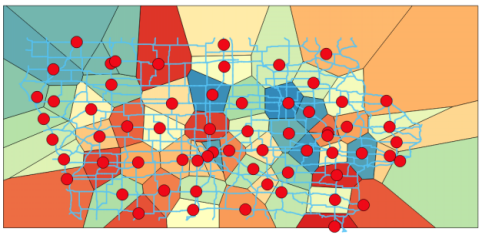


Figure 2: Voronoi Diagram of RWIS Matching

3.4 Derived Features

The original dataset doesn't have any information about a road's degree of curve, number of intersections, and the population density around it. We derive those features by using geometry representation of each road segment and the demographic dataset. Though we can easily extract the length of a road segment, it is hard to determine the degree of curve of the road without real measurements. Instead, we define a reference curve measure as

$$curve_{ref} = \frac{length(r)}{dist(r.start,\ r.end)}$$

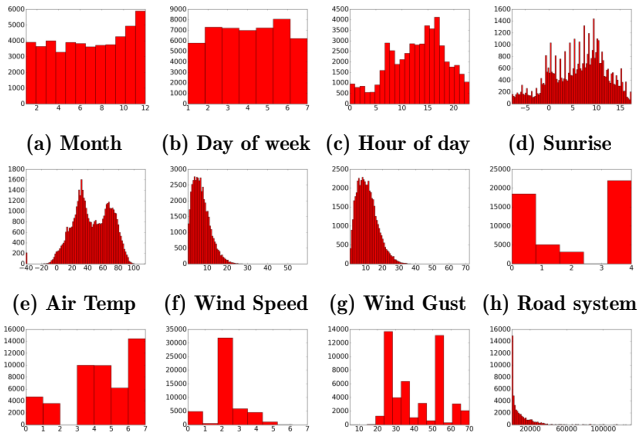
where r is a road segment and $dist(r.start,\ r.end)$ is the length of a straight line between the two ending points of r. Higher ratio of the above reference measure indicates higher degree of road curve.

Also, for each road segment, we count the number of roads that touch it and use this number as its "number of intersections".

Finally, for population density we use the 2010 census data of Iowa to retrieve the population of each census block, and then we compute the population density as

$$density = \frac{\text{population of census block}}{\text{area of the census block}}$$

We add a "population density" feature for each road segment. The value is calculated by using the average population density of census blocks that this road segment passes.



(a) Month (b) Day of week (c) Hour of day (d) Sunrise

(e) Air Temp (f) Wind Speed (g) Wind Gust (h) Road system

(i) Road Class (j) Num of lanes (k) Speed limit (l) AADT

Figure 3: Visualizations of Crash Count vs. Crash-related Features

3.5 Summary of Preprocessed Data

We totally matched 415,153 motor vehicle crashes over 8 years containing 40 features related to traffic crash. We summarize the features as the following four categories:

- **Temporal Factors** include 10 features that describe time information of an incident, e.g. date, hour and other time-derived features, e.g. time to sunrise, ifholiday.
- **Weather Factors** include 5 features that are associated with a specific crash instance, e.g. detailed weather data (e.g. temperature, wind speed) and precipitation.
- **Road Factors** are 7 time-invariant features, which demonstrate the properties of a road segment where an accident happened e.g. road length, road curve and speed limit.

- **Human Factors** includes 18 features that are related to human activities, such as annual average daily traffic and population density.

We also summarize the relations between collected features and accident counts. Figure 3 plots the histograms of selected important features with the horizontal axis as feature values and the vertical axis as accident count. Here, we used dataset of Year 2013 as an example. From Figure 6(a), it shows December months have higher numbers of accidents. Figure 6(c) shows that most of accidents of a day are mainly concentrated within time slot between 15:00 and 18:00, which is the evening rush hours. Figure 3(j) and Figure 3(k) show that roads with 2 lanes or speed limit of 25 mile/h or 55 mile/h have the most accidents of the year. Figure 3(e) reveals the fact that roadway icing increases the risk of traffic accidents around 32 degree in Fahrenheit.

4 PROPOSED APPROACH

In this section, we first present our approaches to feature engineering and sampling, and then describe the classification models. To address spatial heterogeneity, we use the road network connectivity graph to generate a new set of features, to which we refer to as SparitalGraph features and will present the details shortly.

4.1 Feature Engineering

Feature Normalization of continuous features. To mitigate the effect of different magnitudes for different features, we conduct feature normalization on continuous feature (e.g., air temperature, wind speed, annual average daily traffic). We use the Z-normalization method with the formula shown below:

$$x_{norm} = \frac{x - \mu}{\sigma}$$

where μ is the mean value of the original feature and σ is the standard deviation of the feature values.

Handling Categorical Features. Since some classification models (e.g., neural network, Support Vector Machine) are not able to handle categorical features, we convert category features into real values using the one-of-K coding. For example, the feature of Holiday (true or false) is represented by a vector of 2 dimensions with (1,0) representing holiday and (0,1) representing not a holiday.

4.2 Negative Sampling

In order to build a binary classification model to classify a record into accident and non-accident, we need to construct negative examples for training since the collected data are all positive examples (with accidents). Indeed, all road segments at all time points that are not accident points can be treated as negative examples, which can give us a huge pool of negative examples. Nevertheless, using all the negative examples will cause severe imbalance issue. To mitigate this issue, we propose an informative sampling approach that can construct diverse negative samples, with some close to positive examples and with some far from the positive examples. Our approach is that for each positive example we randomly change the value of only one feature among Hour, Day and road ID, and check if the resulting sample is a positive or not, and then add it to the negative pool if it is not a positive example. We choose these three features because the change of them can generate diverse negative examples with differences in the first three categories of features. For each feature, we follow the procedure below:

- **Hour.** If an accident occurred in hour A, we randomly pick another time slot from [0, 24] except for hour A of that day. Changing the hour feature may cause the change of other features related to time in a day (e.g., time to sunrise, sunset, dust, dawn).
- **Day.** If an accident occurred in day B, we randomly pick another day from [1, 365] except for day B of that year. Changing the day feature will cause the change of features related to day, including weather related features and some time related features.
- **Road.** If an accident occurred in road C, then we randomly pick another road segment from all road segments except for road segment C. The change of the road ID will cause the major change in the road specific features and may also cause minor change in other features (e.g., weather related features).

Finally, the data set contains roughly about 3 times negative examples than the positive examples. The exact number of negative samples is 1,245,459.

4.3 SpatialGraph Features

To tackle the spatial heterogeneity, we also take the spatial relationship 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. 4 (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.

There are several ways to tackle the spatial heterogeneity. One way is to divide the roads into different areas according to some criteria (e.g., urban vs rural) and then learn different models for different areas. An issue is that it will reduce the number of training examples, especially the positive examples, for learning each model. The more the divided areas the less the training data for each area. Instead, we propose a different approach to tackle the spatial heterogeneity by inducing a new set of features that consider the spatial relationship between different roads. The idea is to construct a spatial graph between all roads and to conduct the eigen-analysis of the induced Laplacian matrix [2]. We use the resulting top eigen-vectors of the Laplacian matrix as the new features for different roads. 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 [26], 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 4. We vary the number of top eigen-vectors from K = 10 to K = 40 and generate the different number of clusters. Note that the roads in the same colored area are grouped into the same cluster and thus share similar (but not identical) spatialgraph features. The benefit of this approach is that we can vary the number of induced features to capture different levels of spatial heterogeneity in the data.

Algorithm 1: Spectral Clustering

Data: Adjacency Matrix $\boldsymbol{W} \in \mathbb{R}^{n\times n}$, Degree Matrix $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ and number of clusters k

Result: Clusters $A_1, ..., A_k$ for $A_i = \{ j | y_j \in C_i \}$

1 Construct an adjacency matrix $\boldsymbol{W}$ and a degree matrix $\boldsymbol{D}$;

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

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

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

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

6 Cluster the points y_i with KMeans algorithm into cluster $C_1, ..., C_k$;

4.4 Classification Models

We compare four classification models, namely linear Support Vector Machine (SVM) [1], Decision Tree (DT), Random Forest (RF), and Deep Neural Networks (DNN). For SVM, we use the liblinear [11], an efficient library for large-scale data classification, to conduct the experiments, and we tune the C parameter that balances between the training error and the regularization term on a hold-out validatation data. For DT, we use the classification and regression trees (CART) introduced by Leo Breiman [15], which is able to handle both numerical and categorical input features. We use the Python library scikit-learn [21] to train CART with default parameters. To avoid overfitting, we set the maximum tree depth to 13, which gives the best performances according to our tests. For RF, we also use the scikit-learn library and tune the number of trees. We chose 1000 as the number of trees in our experiments since it gives the best result with the available computing resources.

For DNN, we firstly define a fully connected network architecture up to three hidden layers. In the feedforward propagation, we use uniform initializer to initialize network weights between [-0.05, 0.05]. We use rectifier function, denoted as first function of eqn. (1)

$$f(x) = \max(0, x), f(x) = \frac{1}{1+e^{-x}} \quad (1)$$

as activation function for hidden layers. As for output layer, we use sigmoid function to output prediction label of [0,1], denoted as second function of eqn. (1). To define the loss function, we choose logarithmic loss function as

$$-\frac{1}{N}\Sigma_{n=1}^{N}\left[y_i \log\, p_i + \left(1-y_i\right) \log\, \left(1-p_i\right)\right]$$

In back propagation, we use an efficient gradient decent algorithm, Adaptive Moment Estimation (Adam)[12]. It considers the decaying average of previous gradients and previous gradient moments as

$$m_t = \beta_1 m_{t-1} + \left(1-\beta_1\right) g_t,\ v_t = \beta_2 v_{t-1} + \left(1-\beta_2\right) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t},\ \hat{v}_t = \frac{v_t}{1-\beta_2^t},\ \theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t$$

We choose the default settings suggested by the authors that are $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The neural network is trained by Theano [24]. In addition, we apply dropout regularization to prevent overfitting. The final tuned network consists of three hidden layers with number of neurons equal to 120, 160 and 200 with dropout rate of 0.1 and training epoch of 600.

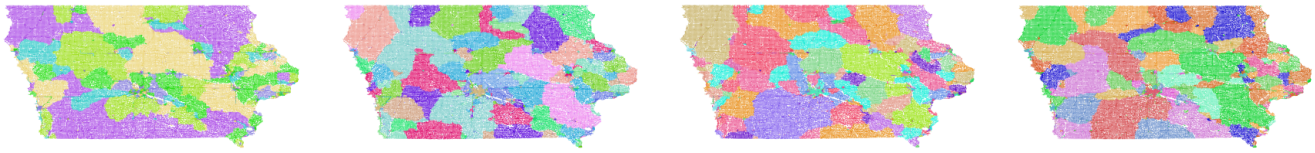
5 EXPERIMENT RESULTS

Experiment Settings: We set up the experiments on Argon High Performance Computing System at the Uinversity of Iowa [16] using a 256GB RAM computing node with 2.6GHz 16-Core CPU. For the training of DNN, we use GPU node on Argon with Nvidia Tesla P100 Accelerator Cards. We use the dataset constructed as introduced in Section 4.1-4.3.

We conduct four experiments: (1) Comparing the results with Informative Negative Sampling and Random Negative Sampling, (2) increasing the number of SpatialGraph features, (3) increasing the training set volume. (4) Training with different groups of features, For the first two experiments, we train the data in year N and test using the data in year N+1 from 2006-2013 and average the performances across these years. For the third experiment, we increase the training set size gradually by adding each year's data at a time. For the last experiment, we use 2012 data for training and 2013 data for testing. To avoid the impact of randomness, each prediction performance measure is the average of three runs, where in each run we sample a new training and testing dataset using the selected sampling strategy.

Evaluation Metrics. We evaluate the models by using the following measures: accuracy, precision, recall, F-Score, AUC (Area Under Curve). The performance of the four predictive models are summarized in Table 1. A simple overview of this table reveals that RF, DT, DNN perform consistently better than linear SVM with varying number of spatialGraph features, **which verifies that linear model is not effective for predicting the traffic accident.** We analyze more detailed results below.

[1] Although a non-linear SVM model can be also learned by using kernel tricks, however, due to the long training time of kernel SVM on our big data set, we do not conduct the experiments using kernel SVM.



(a) K = 10 (b) K = 20 (c) K = 30 (d) K = 40

Figure 4: Clustering Maps of the Road Network with Different Number of Clusters

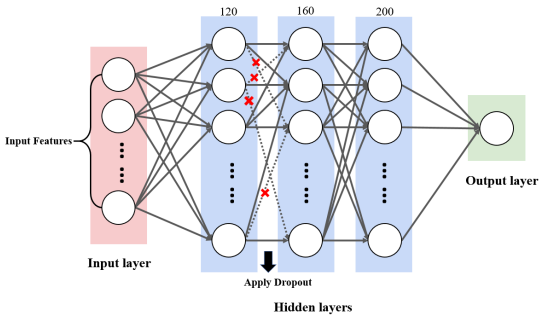


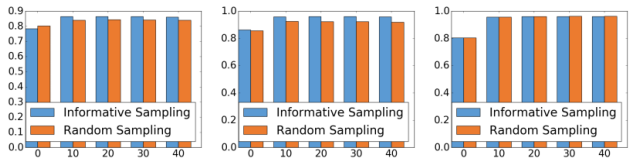
Figure 5: A Three-layer Deep Neural Network Architecture

Table 1: Comparison of Different Classification Models

	K[4]	AUC	Recall	Precision	F1	Accuracy
SVM[1]	0	0.5471	0.3305	0.2894	0.3066	0.6307
	10	0.5549	0.3276	0.3042	0.3131	0.6444
	20	0.5507	0.3133	0.3029	0.3059	0.6482
	30	0.5543	0.3358	0.3044	0.3175	0.6419
	40	0.5538	0.3434	0.3074	0.3221	0.6411
DT[2]	0	0.7847	0.3506	0.6687	0.4590	0.7940
	10	0.8636	0.5735	0.9275	0.7079	0.8827
	20	0.8646	0.5762	0.9271	0.7100	0.8832
	30	0.8640	0.5781	0.9273	0.7114	0.8837
	40	0.8607	0.5765	0.9263	0.7098	0.8832
RF[3]	0	0.8623	0.3728	0.8779	0.5230	0.8310
	10	0.9585	0.5963	0.9875	0.7431	0.8976
	20	0.9597	0.5821	0.9903	0.7326	0.8946
	30	0.9612	0.5873	0.9904	0.7366	0.8958
	40	0.9592	0.5635	0.9919	0.7179	0.8902
DNN	0	0.8036	0.5058	0.6135	0.5540	0.7974
	10	0.9575	0.8523	0.9221	0.8858	0.9453
	20	0.9608	0.8663	0.9280	0.8961	0.9500
	30	0.9612	0.8685	0.9309	0.8986	0.9512
	40	0.9603	0.8689	0.9302	0.8985	0.9511

[1] C=100, [2] Max depth=13, [3] Num of Trees=1000
[4] K represents the number of SpaitialGraph features

Informative Sampling vs Random Sampling. We compare our proposed informative sampling method with random sampling. We feed two training sets to each model generated with informative sampling and random sampling, respectively. Figure 6 shows the results obtained on three different models, where blue and red bars represent the proposed Informative Sampling and Random Sampling (x axis: K, number of spatialGraph features, y axis: AUC), respectively. The results indicate that our proposed sampling method performs better than random sampling on decision tree and random forest (yielding 0.013 and 0.031 increase, respectively). In Neural Network, the two sampling methods have very similar performances with less than 0.001 difference. Due to the poor performance of linear SVM model, we didn't further evaluate it. For the following experiments, we will use our proposed informative sampling method to evaluate the models.

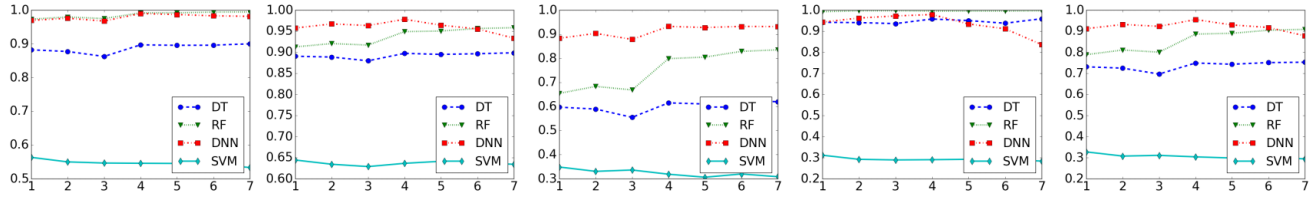


(a) DT (b) RF (c) NN

Figure 6: AUC Comparisons Over Three Models

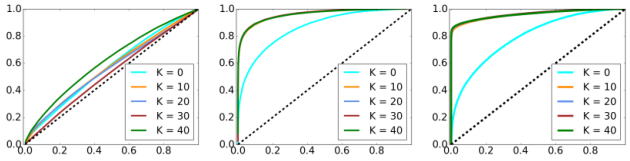
Effect of SpatialGraph Features. We train each model with different numbers of SpatialGraph features by varying the value of K. Note that $K = 0$ corresponds to the case where no SpatialGraph features is used. From the results in Table 1, we can observe that incorporating the SpatialGraph features can significantly improve the performance especially for the non-linear models, **which verifies that modeling the spatial heterogeneity is very important for traffic accident prediction.** In addition, we can observe that DT, RF, and DNN generally achieve the best performance in all the measures when $K = 30$. The performance improvement when K>10 are marginal. An explanation to this observation is that: as the number of spatialGraph features grow, the data sparsity issue becomes more severer, and start to lower the prediction performances. Figure 8 shows the ROC curves with different K for each model. As can be observed, the curves move towards the left-top corner when increasing K, while the improvement becomes marginal when K>10 for three models.

Effect of Training Data Size. Next, we verify the effect of training data volume on traffic accident prediction. To this end, we vary the size of the training data. We choose the first N year's data as the training set and use data from year $N + 1$ as the testing set, where N is varied from 1 to 7. The results are plotted in Figure 7, from which we can see that more training data generally leads to the better performance, specifically, improving the recall and AUC for most models. However, for DNN, as the increasing of training data size, accuracy, precision and F1 decrease in different degree. This



(a) AUC (b) Accuracy (c) Recall (d) Precision (e) F1

Figure 7: Evaluations on Different Size of Training Examples When Choosing Best K for Each Model



(a) Linear SVM (b) Random Forest (c) Neural Network

Figure 8: ROC Curves of Three Models

might be caused by overfitting since the parameters of DNN are tuned by using one year's data. Overall, varying the training data size doesn't effect the performance of SVM.

Effect of the Different Category Features. Finally, we investigate the effect of different features on model prediction by removing one category features at each time when choosing the best K for each model. The results are presented in table 2. We can observe that removing road-related features has dramatically impacted on DT and RF. However, removing the other three categories has slight impact on three models. DNN is robust to changes in features.

Table 2: Comparison of Different Different Category Features

	R[1]	AUC	Recall	Precision	F1	Accuracy
DT	H	0.8700	0.5977	0.9302	0.7278	0.8887
	T	0.8770	0.6220	0.9247	0.7437	0.8933
	R	0.5717	0.0876	0.2785	0.1333	0.7163
	W	0.8729	0.6105	0.9227	0.7348	0.8903
	N	0.8717	0.6094	0.9210	0.7335	0.8897
RF	H	0.9633	0.6305	0.9917	0.7709	0.9067
	T	0.9781	0.7102	0.9894	0.8269	0.9260
	R	0.6127	0.0176	0.4121	0.0338	0.7491
	W	0.9679	0.6236	0.9911	0.7655	0.9049
	N	0.9625	0.5898	0.9921	0.7398	0.8967
DNN	H	0.9609	0.8707	0.9388	0.9035	0.9537
	T	0.9691	0.8935	0.9540	0.9228	0.9628
	R	0.9583	0.8681	0.9294	0.8976	0.9507
	W	0.9609	0.8741	0.9262	0.8993	0.9513
	N	0.9608	0.8696	0.9388	0.9029	0.9534

[1] R represents the removed category features: H - Human, R - Road, W - Weather, T - Temporal, N - None

Summary. The overall performance of RF and DNN are comparable and much better than that of SVM and DT. Specifically, DNN achieves the highest AUC and accuracy of 0.9612 and 0.9512. The above experiment results show that incorporating spatial structure of the road network properly could significantly improve prediction performance, and increasing the training data volume can generally improve the measures, especially recall and AUC. However, using more training data and spatialGraph features might lead to other issues such as data sparsity and overfitting. These issues need to be considered when setting the parameters.

Discussion. The weather-related features could not be obtained in the real-time prediction. However, the current weather forecast technique is highly accurate for short term prediction, according to National Weather Service data [22]. Thus, our approach still works effectively.

6 CONCLUSION AND FUTURE WORK

This paper investigated the problem of traffic accident prediction using heterogeneous urban data, an importance problem to transportation and public safety. This problem is also very challenging due to imbalanced classes, spatial heterogeneity and its non-linear separable nature. Prior research on this topic mostly used simply used classical data mining tools or models on limited amount of data, without addressing spatial heterogeneity and class imbalance properly. In this paper we formulated the problem as a binary classification problem. We obtained and map-matched fine-grained datasets such as all the motor vehicle crashes in Iowa from 2006 to 2013, detailed road network, and hourly weather data, and evaluated the performance of several classification models. We also incorporated road network connectivity relationship into the classification process through Eigen analysis. Results show that our proposed approach significantly improved (DNN) accuracy and AUC to 0.9512 and 0.9612, respectively.

For future work we would like to explore AUC optimization techniques as well as online learning methods to predict traffic accidents in real-time. We also plan to investigate approaches to predict the precise number of accidents.

ACKNOWLEDGMENT

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.

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] Mikhail Belkin and Partha Niyogi. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, Vol. 14. 585–591.

[3] 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.

[4] Ciro Caliendo, Maurizio Guida, and Alessandra Parisi. 2007. A crash-prediction model for multilane roads. *Accident Analysis & Prevention* 39, 4 (2007), 657–670.

[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] 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*.

[8] Miao M Chong, Ajith Abraham, and Marcin Paprzycki. 2004. Traffic accident analysis using decision trees and neural networks. *arXiv preprint cs/0405050* (2004).

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

[10] Minh X. Hoang, Yu Zheng, and Ambuj K. Singh. 2016. FCCF: forecasting citywide crowd flows based on big data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2016, Burlingame, California, USA, October 31 - November 3, 2016*. 6:1–6:10. https://doi.org/10.1145/2996913.2996934

[11] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. 2003. A practical guide to support vector classification. (2003).

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

[13] 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.

[14] 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.

[15] Wei-Yin Loh. 2011. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1, 1 (2011), 14–23.

[16] University of Iowa. 2016. Neon Overview and Quick Start Guide. (2016). https://wiki.uiowa.edu/display/hpcdocs/Neon+Overview+and+Quick+Start+Guide

[17] Department of Transportation. 2016. Iowa DOT GIS REST Services: Crash Data. (2016). https://gis.iowadot.gov/public/rest/services/Traffic_Safety

[18] Department of Transportation. 2016. Iowa DOT GIS REST Services: Road Network. (2016). https://gis.iowadot.gov/public/rest/services/RAMS

[19] Department of Transportation. 2016. The Iowa Environmental Mesonet: Roadway Weather Information System. (2016). https://mesonet.agron.iastate.edu/RWIS/

[20] Jutaek Oh, Simon P Washington, and Doohee Nam. 2006. Accident prediction model for railway-highway interfaces. *Accident Analysis & Prevention* 38, 2 (2006), 346–356.

[21] 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.

[22] National Weather Service. 2017. NATIONAL DIGITAL FORECAST DATABASE. (2017). https://sats.nws.noaa.gov/~verification/ndfd/

[23] 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.

[24] Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* abs/1605.02688 (May 2016). http://arxiv.org/abs/1605.02688

[25] 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.

[26] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17, 4 (2007), 395–416.

[27] Colin Ware, William Knight, and David Wells. 1991. Memory intensive statistical algorithms for multibeam bathymetric data. *Computers & Geosciences* 17, 7 (1991), 985–993.

[28] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence*.