Incident Analysis and Prediction Using Clustering And Bayesian Network

Geoffrey Pettet, Saideep Nannapaneni, Benjamin Stadnick, Abhishek Dubey, Gautam Biswas
School of Engineering, Vanderbilt University, Nashville, TN 37235, USA

***Abstract*—Advances in data collection and storage infrastructure offer an unprecedented opportunity to integrate both data and emergency resources in a city into a dynamic learning system that can anticipate and rapidly respond to heterogeneous incidents. In this paper, we describe integration methods for spatio-temporal incident forecasting using previously collected vehicular accident data provided to us by the Nashville Fire Department. The literature provides several techniques that focus on analyzing features and predicting accidents for specific situations (specific intersections in a city, or certain segments of a freeway, for example), but these models break down when applied to a large, general area consisting of many road and intersection types and other factors like weather conditions. We use Similarity Based Agglomerative Clustering (SBAC) analysis to categorize incidents to account for these variables. Thereafter, we use survival analysis to learn the likelihood of incidents per cluster. The mapping of the clusters to the spatial locations is achieved using a Bayesian network. The prediction methods we have developed lay the foundation for future work on an optimal emergency vehicle allocation and dispatch system in Nashville.**

I. Introduction

Emerging Trends: The advancement in sensors and information transfer technologies provide opportunities to collect large amounts of data on complex operations and processes that govern various aspects of city life. In such situations, where it is an almost intractable task to build complex models, data driven approaches can produce more informed solutions to problems than using heuristics and ad-hoc approaches. Such data driven approaches have been successfully applied in a number of areas, ranging from monitoring industrial processes [1] to identifying students at risk of emotional disorders [2]. Today, advances in data collection (such as wireless sensor networks [3]) and storage infrastructure (such as distributed hash rings [4]) have allowed information to be collected and analyzed for applications not possible before, such as urban analytics [5] and emergency response services.

In this paper, we describe a toolchain that will enable fire departments to analyze multiple, distributed incident occurrences that they must respond to. Our goal is to analyze historical incident data and develop predictive models that can help the department efficiently allocate and route emergency vehicles to incidents as they occur over a large, distributed area. Minimizing response times increases victim survival rates [6] and frees vehicles to respond to other incidents more quickly. Any such optimal dispatch algorithm is typically based on a sequential optimization that requires prediction of the likelihood of future incidents occurring in a given area, so it can plan ahead.

Incident prediction using the negative binomial distribution [7], artificial neural networks [8], and hierarchical analysis [9] have been used to great effect when attempting to predict incident frequency for specific areas, and have helped determine features of roadways that affect incident occurrence. Prediction methods such as the negative binomial regression [10] and random effect probit models [11] have also been used to analyze feature effects on accident frequency, and generating predictive models for specific areas. Unfortunately, these studies generally make assumptions about the locations that they are analyzing. For example, they study a specific length of freeway, or look at only intersections and their features in a specific city. To create a predictive model for an entire metro area however, location agnostic features such as weather and time must be considered for the incidents.

Contributions: This paper describes a toolchain to forecast the likelihood of incidents, specially motor vehicle incidents, occurring in a large, geographical area. This paper describes major components of our prediction and analysis toolchain:

1) An unsupervised clustering approach for grouping incidents with similar characteristics. We hypothesize that incidents within each group will have similar arrival times, making forecasting for each group more accurate. This is validated by our results described in Table III-E: the average log-likelihood of cluster survival model accuracy is significantly higher (-16,100.8) than models built from the entire dataset (-180,243.8).
2) Predictive models for each cluster using survival analysis.
3) A mapping of these cluster predictive models to spacial locations using a Bayesian network.
4) Compose all of the data preprocessing, analysis, and prediction routines in the form of a toolchain to facilitate analysis of data received from the Nashville fire department from February 2014 to February 2016, and then validate our toolchain on data from February 2016 to December 2016. This toolchain will facilitate future emergency vehicle dispatch optimization analysis.

Paper Outline Section II presents prior work on incident prediction. Section III-A describes the data used in our case study for Nashville Motor Vehicle Accident (MVA) response dispatching. In section III-B we formally describe the problem specification and then detail each step in the prediction toolchain, and simultaneously present the results of the case study. Section IV presents a discussion, and Section V presents the conclusions of the paper.

978-1-5386-0435-9/17/$31.00 ©2017 IEEE

II. Related Research

Vehicle accident analysis has been an important area of research due to their large safety and monetary costs and their impact on human life. Miaou and Lum found that due to over dispersion often present in accident data, the more general Negative Binomial distribution is often superior to the Poisson regression in this area [7]. Abdel-Aty and Radwan applied the negative binomial technique to model accident frequency for a principal arterial in Central Florida [12]. Using data from 1606 accidents over three years, they found eight features to be significant in determining the frequency of accidents, including segment length, shoulder width, and annual average daily traffic (AADT). Ackaah and Salifu used negative binomial regression to model 76 rural highways in Ghana [10]. They found that the negative binomial distribution fit their data reasonably well, and that five features (including traffic flow and road segment length) were significant in determining accident frequency.

Chin and Quddus built on the research by suggesting that the random effect negative binomial (RENB) model is superior to the negative binomial model [13]. They reasoned that the negative binomial's assumption that accident data is uncorrelated in time is inappropriate for accidents, due to serial correlation in the accident data. The RENB model accounts for temporal effects by treating the data in a time-series cross-section panel. Their model, based on signalized intersections in Singapore, found that eleven features had a significant impact on accident frequency, and improved on the model found with the negative binomial model.

Chang explored artificial neural networks (ANN) as another alternative to the negative binomial regression model [8]. The negative binomial model assumes a predefined underlying relationship between the dependent and independent variables. If this assumption is violated, it will lead to erroneous accident estimations. ANN avoids this by making no assumptions regarding the variable's relationship. Raut and Karmore have combined an Artificial Neural Network to analysis a large amount of input data with a fuzzy logic system that uses this data to predict accident severity [14]. Unfortunately this technique requires large amounts of external traffic information that may not be available to dispatch services.

Researchers have also explored applying various classification methods to traffic problems. For example, Moreira-Matias [15] uses boosted decision trees to classify traffic jam events. Unfortunately these methods do not apply to our problem, since we are looking for the probability that an accident will occur in each location, not to classify accidents to a location.

Survival analysis has also been widely used and is the one of the core components of our toolchain. This analysis has been applied by researchers to predict accident duration. Chung applied the log-logistic accelerated failure time model to 2 years of Korean freeway accident duration data [16]. Using eight features to characterize accidents, they found that the model provided reasonable duration prediction based on the mean absolute percentage error scale. Kang and Fang (2011) similarly applied the Weibull prediction model to 3 years of Jiaxing city's freeway incident data to find predictive factors affecting incident duration, and found the model to be reasonable.

The research presented thus far makes strong assumptions about the location and/or type of accident during analysis. For example, they might only look at accidents occurring at intersections or along a single stretch of highway. While these assumptions are helpful for small scale or specific analysis, they make generalizing the results difficult. In the past, our group has studied the accident prediction problem, focusing on spatial grids [17]. However, in this paper we explore a generalized, yet unsupervised approach to categorizing incidents. Therefore, the method presented in this paper clusters on *individual incidents* rather than *grid sections*. This leads to tighter, more meaningful clusters for dispatch allocation, which is shown by our improved likelihood results shown in the Table III-E. This is helpful since an important aspect of emergency response is ensuring that appropriate equipment is dispatched based on the incident type.

III. Our Approach

A. Data Specification

The majority of the data used in this study was provided by the Metro Nashville Fire Department in the form of a relational database scrubbed to eliminate personally identifiable information. The database contained approximately two years of incidents occurring from February 2014 to February 2016. In total, there were 477,837 unique incidents recorded in the database, the majority of which occurred in Davidson County, Nashville, Tennessee.

Motor vehicle accidents were extracted from the database according to the following criteria: the location of the incident was fully specified by GPS coordinates, the incident occurrence time was known, the first unit arrival time that occurred after the incident occurrence time was also known, and the incident was classified by emergency medical dispatch card numbers starting with 29 to ensure it was a motor vehicle accident. Using these criteria 19,910 motor vehicle accidents were extracted for the clustering algorithm.

In addition to the information obtained from the database, weather condition information and information regarding the type of road on which the accident occurred was obtained from DarkSky and OpenStreetMaps, respectively. Using these three sources of information, the features described in the clustering analysis subsection III-C were extracted for clustering.

B. Overview of the Approach

The toolchain described in this paper consists of 4 major components as shown in figure 1: data preprocessing, clustering incidents (modeled in figure 2), learning survival models for each incident cluster, and mapping these predictive models to locations (Both shown in figure 3). Each step is explained in detail in the following sections.

For prediction, we assume that the city is divided into a grid of regularly sized half mile hexagonal sections, which are referred to as **hex cells** in the remainder of the paper. For each hex cell, we have data regarding incidents that took place in that grid in the given period. We also assume that the incidents

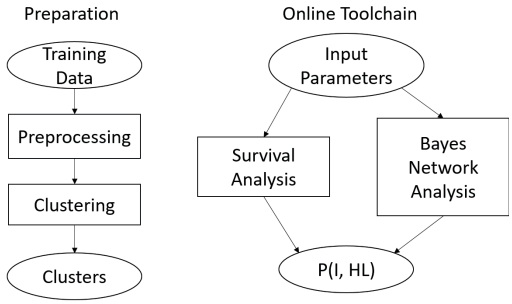

Fig. 1. Toolchain Block Diagram. P(I, HL) refers to the joint probability of an incident occurring in a particular hex cell

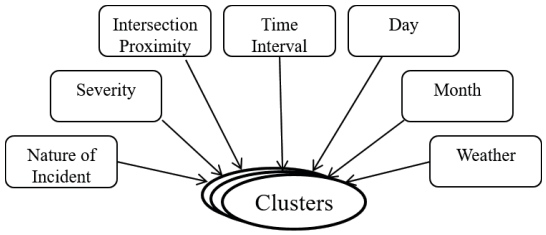

Fig. 2. Cluster Generation Model

in each cell are independent of incidents in other cells. This information enables us to train the Bayesian network that we describe later in the paper.

C. Clustering Analysis

The first step of the toolchain is clustering the incidents into similar groups. There are three primary steps to this process: (1) choosing incident features to cluster on, (2) calculating the similarity values between each pair of incidents and (3) running a hierarchical clustering algorithm, SBAC [18] to generate a dendrogram, and then establishing a minimum distance criterion as separation among clustering, and therefore, establishing the number of clusters that make up the dataset. forming a dendrogram from these values, and cutting that dendrogram to form the optimal number of clusters.

Incident Feature Selection: Table I describes the features we chose to categorize the incidents. We discretized the continuous values of time and the distance to the nearest intersection to reflect trends in the data.

Similarity Calculation: Once features of interest are chosen for the incidents, they are used to calculate a similarity measure between incident pairs. Traditional clustering methodologies generally focus on either numeric valued data [19] (k-means clustering [20], for example) or nominal valued data (known as conceptual clustering [21]), but are not designed for mixed numeric and nominal data. While the features in our case study are all nominal, other applications of this toolchain may require some numeric features to be considered. For this reason, we use a similarity measure that works well with mixed typed data.

Specifically, we use a similarity measure created by Li and Biswas that has been shown to work well for mixed data types [18]. Their method, which is a generalized version of a measure proposed by Goodall for biological taxonomy [22], uses unusual characteristics shared by data objects to

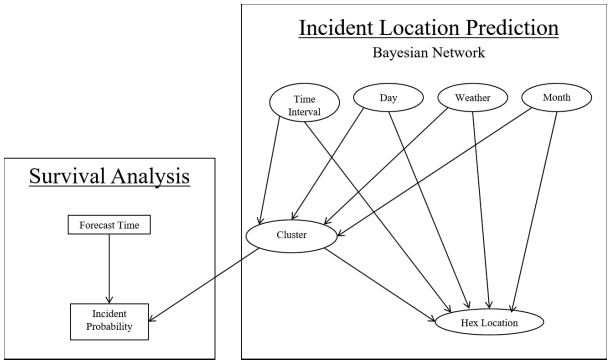

Fig. 3. Prediction Toolchain Model

TABLE I
INCIDENT FEATURES CONSIDERED

Feature	Description	Source
Road type	Type of road incident took place on, such as freeway or primary	Obtained from OpenStreetMaps based on the GPS coordinates and street address
Weather	Weather conditions at the time of the incident	Obtained from DarkSky based off GPS coordinates and time of incident
Severity	Severity measure based off Fire Department codes, with 'A' as least severe to 'D' being most severe	Obtained from the incident's associated emergency medical dispatch card number
Nature of Accident	Description of incident	Obtained from emergency medical dispatch card number
Time of Day	time in which incident occurred: early morning, late morning, afternoon, or night	Obtained from Nashville fire department data
Day	The day of the week the incident occurred on	Obtained from Nashville fire department data
Month	The month in which the incident occurred	Obtained from Nashville fire department data
Intersection Proximity	How close the incident occurred to an intersection: On, near, or far from the intersection	Obtained from OpenStreetMaps based on the GPS coordinates of the incident

determine their similarity. Specifically, "a pair of objects *(i,j)* is considered more similar than a second pair of objects *(l,m)* if *i* and *j* exhibit a greater match in feature values that are less common in the population. In other words, similarity among objects is decided by the uncommonality of their feature value matches" [18]. This helps create tight clusters likely to share unique feature values.

To demonstrate how to calculate the similarity of a pair of objects using this method, it is helpful to use a toy example. Consider the objects described in Table II: there are 6 ball objects, each of which have a color (a nominal feature) and a weight (a numerical feature).

To calculate the similarity of a pair of objects, the first step is to determine the individual feature similarities - the color and weight, in this case. The technique used to calculate this depends on if the feature is nominal or numeric. These individual feature similarities are then combined into a total similarity for the pair of objects. The techniques for each of these steps are described below.

Nominal Feature Similarity: Let us first consider nominal

TABLE II
TOY EXAMPLE FOR SIMILARITY CALCULATION

Ball ID	Color (Nominal)	Weight (Numeric)
Ball 1	Red	15.0 kg
Ball 2	Red	10.0 kg
Ball 3	Red	10.0 kg
Ball 4	Yellow	10.0 kg
Ball 5	Yellow	7.5 kg
Ball 6	Blue	5.0 kg

feature similarity. There are two cases for nominal features: the two objects have either the same feature value, or different feature values. If the two objects' feature values are not equivalent their similarity score is 0 (i.e. they are not similar at all), but if the values are the same then the similarity score is somewhere between 0 and 1. Applied to our toy example, Balls 1 and 4 have 0 similarity for the Color feature since red and yellow are different colors, and there is no way to relate the two. Balls 1 and 2, on the other hand, have some non-zero similarity for Color since they are both red.

The exact feature similarity score when the two feature values are equivalent is a function of that value's uncommonality in the population: the more common a feature value, the less similar any pairs with that value are considered to be. For example, yellow balls are considered more similar than red balls in our toy dataset, since there are more red balls.

This information is used to create the object pair's *More Similar Feature Value Set (MSFVS($(v_i)_k$)*, which is the set of all values for nominal feature *k* that are equally or more similar than the object pair's feature value. For example, the *MSFVS* for balls 1 and 2 would be {Red, Yellow} since yellow is rarer than red (blue is omitted, as there are no pairs with that color). The *MSFVS* for balls 4 and 5, however, would only be {yellow}, since yellow is the rarest color represented by a pair of ball objects.

Equation 1 shows how the *MSFVS* is used to calculate the similarity score between the two objects:

$$(S_{ii})_k = 1 - (D_{ii})_k = 1 - \sum_{l \in MSFVS((V_i)_k)} (p_l)_k^2 \quad (1)$$

where $(p_l)_k^2)$ is the probability of picking a pair $((V_i)_k, (V_i)_k) \in MSFVS((V_i)_k)$ for feature *k* at random, and $(D_{ii})_k$ is the dissimilarity of the objects.

Let's apply this to balls 1 and 2 in our example. As discussed earlier, the balls both have the Color red, and their MSFVS is {Red, Yellow}. Given this set, the similarity for the Color value of red is $S_{(ball1,ball2)})_{Red} = 1 - (D_{(ball1,ball2)})_{Red} = 1 - (p_{red}^2 + p_{yellow}^2) = 0.733$. This means that the color red contributes 0.733 to the similarity score between balls 1 and 2.

Numeric Feature Similarity: The method for calculating the similarity for numeric features is slightly different from nominal features. When feature values are not equivalent, their similarity is determined in the traditional manner: one pair of objects is more similar than another if their feature values are closer together. The weights of balls 5 and 6 are considered more similar than 4 and 6, for example, since the difference in their weights is smaller. It is only when two pairs of values have equivalent differences that the uniqueness of the values is considered.

This information is used to create a pair of value's *More Similar Feature Segment Set (MSFSS($(V_i)_k, (V_j)_k$))*, which includes all pairs of values that are more similar due to the criteria described above.

Similar to nominal features, this is used to calculate the feature value similarity as follows: the probability of picking two objects from the population having values $(V_l)_k$ and $(V_m)_k$ for feature *k* where $((V_l)_k, (V_m)_k) \in MSFSS((V_i)_k, (V_j)_k)$ is

$$\alpha_{lm} = \begin{cases} 2(p_l)_k(p_m)_k = \frac{2(f_l)_k(f_m)_k}{n(n-1)}, (p_l)_k \neq (p_m)_k \\ (p_l)_k(p_m)_k = \frac{(f_l)_k((f_l)_k-1)}{n(n-1)}, (p_l)_k = (p_m)_k \end{cases} \quad (2)$$

where f_l and f_m are the frequency of values $(V_l)_k$ and $(V_m)_k$ and *n* is the total number of objects in the population. The similarity is then computed as in equation 3:

$$(S_{ij})_k = 1 - (D_{ij})_k = 1 - \sum_{l,m \in MSFSS((V_i)_k,(V_j)_k)} \sigma_{lm} \quad (3)$$

where σ_{lm} is the appropriate probability distribution from equation 2 and $(D_{ij})_k$ is the dissimilarities of the two objects. For example, take balls 1 and 2 again. The MSFSS for their value segment (10.0, 15.0) is {(5.0, 7.5), (7.5, 10.0), (10.0, 10.0), (10.0, 15.0)}. Notice that even though balls 6 and 4's values (5.0, 10.0) are as close together as (10.0, 15.0), they are not included in the set. This is because there are more balls with weights in the range [5.0, 10.0] then [10.0, 15.0], making the weights of balls 5 and 6 more unique (and therefore more similar).

This makes the ball's similarity $(S_{(b1,b2)})_w = 1 - (D_{(b1,b2)})_w = 1 - (2p_{5.0}p_{7.5} + 2p_{7.5}p_{10.0} + p_{10.0}p_{10.0} + 2p_{10.0}p_{15.0}) = 0.333$. This means that the weight feature contributes 0.333 to balls 1 and 2's similarity.

Total Object Similarity Aggregation: To aggregate these individual feature similarities for a pair of objects into the total similarities, we apply Fisher's χ^2 transformation [23] to numeric features, assuming that individual results are expressed as the square of a standard normal deviate. Continuing our simple example with balls 1 and 2, their aggregate numeric similarity would be $(\chi_c)^2_{(Ball1,Ball2)} = -2(ln(0.667) = 0.8109$.

For nominal features, Lancaster's mean value χ^2 transformation [24] is applied, as the traditional Fisher's transformation has been shown to cause deviations in the mean and standard deviation when applied to features with a small number of possible observations [24]. The nominal aggregate for balls 1 and 2 is therefore $(\chi_d)^2_{(Ball1,Ball2)} = 2(1 - \frac{(0.267 * ln(0.267)) - 0}{0.267 - 0}) = 4.641$

These two χ^2 distributions can be added to determine the aggregate χ^2_{agg} distribution. The significance value of this distribution gives the aggregate dissimilarity of the two objects, which can be looked up in standard tables. This makes the final dissimilarity for our example balls 1 and 2 approximately 0.1, giving them a similarity of 0.9 in this population.

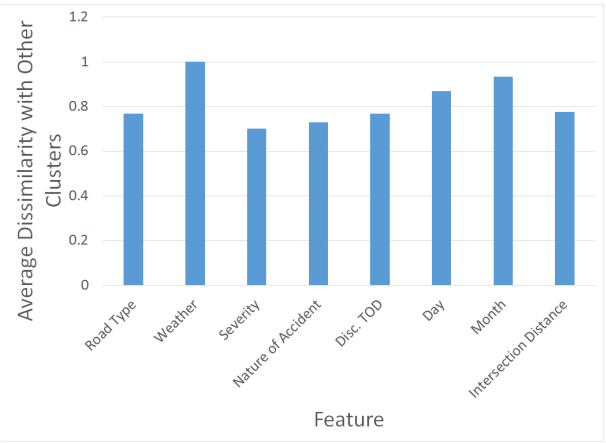

Fig. 4. Cluster 7 - Average Feature Dissimilarity

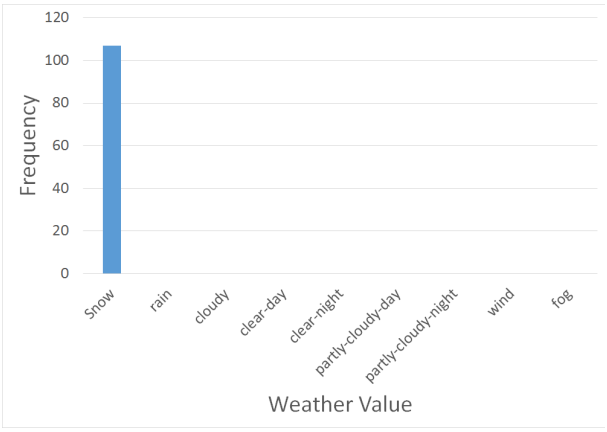

Fig. 5. Cluster 7 - Weather Feature Values

The result of performing this analysis on each pair of objects is a *Dissimilarity Matrix* defining the similarity relation between each pair. We then use these dissimilarities to perform *agglomerative hierarchical clustering*. Hierarchical clustering algorithms construct a *dendrogram*, which is a hierarchy of possible clusters. In agglomerative clustering, these groups are built from the bottom up: each data point starts in its own leaf group. During each iteration of the algorithm, the most similar pair of groups are merged into one group at the next level. This constructs a tree from the bottom up, and continues until there is only one root group containing all incidents [25]. We now determine the optimal level to cut this dendrogram.

Establishing the number of Clusters: To determine the optimal level to cut the dendrogram we score each possible clustering with a weighted silhouette value. Silhouette analysis compares each incident's similarity with its assigned cluster to its similarity to the next most similar cluster [26]. Formally, for each object *i*, let *a(i)* be the average dissimilarity of *i* with all data within the same cluster, and *b(i)* the lowest average dissimilarity of *i* to any cluster of which *i* isn't a member. The silhouette value of *i* is:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{4}$$

which produces silhouette values in the range $-1 \leq s(i) \leq 1$, where a high value indicates that the incident is well matched with its assigned cluster, while a low value indicates it is more similar to objects in its neighboring cluster. Finding the average silhouette score across all objects shows how well the objects are clustered in general:

$$\frac{1}{n} \sum_{i \in dataObjects} s(i), \tag{5}$$

where n is the total number of objects being clustered. To lower the complexity of our groupings, we augment traditional silhouette analysis with a complexity weight, favoring cuts with fewer clusters:

$$w \cdot m + (\frac{1}{n} \sum_{i \in dataObjects} s(i)), \tag{6}$$

where the weight *w* is multiplied by the number of clusters in the current cut *m*.

Applying this complexity-weighted silhouette scoring to groupings produced by the dendrogram helps us find the optimal number of clusters. *By applying this to our data, we found the optimal number of clusters to be 13.*

Individual Cluster Analysis: Each of the clusters can be analyzed to determine the characteristics found to be unique to that cluster by the Similarity Based Agglomerative Clustering (SBAC) algorithm described earlier. For some clusters, this characterization is simple to visualize using feature dissimilarity - the more dissimilar a feature, the more unique it is.

Take our 7th cluster, for example. When examining the average feature dissimilarities for the cluster in figure 4, we see that the weather values are most unique. The weather represented in figure 5 shows this cluster is dominated by incidents that occurred in snowy conditions. This coincides with our goal, as it makes sense that incidents occur at a different rate in snowy weather.

D. Survival Analysis per Cluster

The next step of the toolchain is to create predictive models for each group. A group of statistical methods that are particularly well suited to this is survival analysis, which analyze if and when an event of interest is likely to take place [27]. More precisely, "the analysis of data that correspond to the time from a well-defined *time origin* until the occurrence of some particular event or *end-point*" [28]. In this case, the *time origin* is the current time, and the event whose occurrence we are interested in is an incident. In particular, we use an accelerated failure time model, which regresses the logarithm of the survival time over the covariates [29].

Formally, the survival function is defined as $S(t) = 1 - F_t(t)$ where $F_t(t)$ is the cumulative distribution function of the arrival time variable T. To model our survival function, we use an exponential distribution for our regression due to its memoryless property: the predicted time to the next event does not depend on the elapsed time since the last event [30]. We hypothesize that an incident's arrival time does not generally depend on any past incidents, making this property desirable and used in other motor incident studies [31].

TABLE III
COMPARISON OF CLUSTER VS. NON-CLUSTERED PREDICTION

Cluster	Log-Likelihood
1	-89,780.2
2	-52,030.0
3	-5,041.5
4	-15,694.4
5	-11,912.6
6	-22,706.9
7	-1,788.1
8	-719.7
9	-957.5
10	-763.5
11	-164.4
12	-316.9
13	-7,434.8

We applied this regression analysis to each cluster's incident data to learn their predictive models. For comparison, we also applied predictive models to the entire dataset: the same survival analysis as well as the popular negative binomial analysis discussed earlier in the paper. We compare the model's accuracy using the log-likelihood scale: the likelihood of a model is the probability of some observed values (the past data, in this case) occurring given said model and its parameters. The natural logarithm of this likelihood is known as log likelihood, and easier to handle mathematically. By comparing two model's log likelihoods, we can determine which model better fits the historical data (as it will have a high log-likelihood value).

The Log likelihoods of each cluster's survival models are shown in table III-D, while the comparisons are shown in table III-E. The average log likelihood for the 13 clusters was **-16,100.8**, compared to the values of **-180,243.8** and **-178,488.9** of the survival model and negative binomial model applied to the entire dataset, respectively. As we are attempting to maximize log likelihood, our clusters' models showed to be an order of magnitude more accurate than models applied to the entire dataset. This reinforces our hypothesis that similar incidents have similar arrival rates.

E. Bayesian Network Analysis for Associating Clusters with Hex Cells

The last step to the prediction toolchain is using the newly created survival models for each cluster to determine the likelihood of an incident occurring in a particular hex cell. To be able to predict the most likely hex cell, we first learn the distribution of incidents pertaining to each cluster across the cells conditioned on the features shown in Fig. 3 (Time interval, Day, Weather and Month). As these features are categorical in nature, we learn the conditional probability distribution of hex cells, represented as $P(HL|T, D, W, M, C)$ for every combination of the incident features. Here, HL, T, D, W, M, C refer to Hex Location, Time Interval, Day, Weather, Month and Cluster respectively. In the Nashville Fire Incident dataset, the number of levels (distinct possible values) for these incident features are 4, 7, 10 and 12 respectively, totaling to 3360 distinct combinations. After learning the conditional relationships in the Bayesian network (Fig. 3), we use it to find the probabilities of incidents in each location. The systematic procedure for incident procedure is given below.

TABLE IV
COMPARISON OF CLUSTER VS. NON-CLUSTERED PREDICTION

Method	Log-Likelihood
Average of Cluster Survival Models	-16,100.8
Entire Dataset - Survival Model	-180,243.8
Entire Dataset - Neg. Binomial	-178,488.9

1) Cluster Identification: The probability of choosing a particular cluster $P(C|T, D, W, M)$ is based on current time and environment parameters: for example, cluster 7 described in section III-C would become much more likely in snowy weather than clear conditions.
2) Survival probability: We then determine the probability of an incident occurring of this type via the cluster's survival model $P(I|C, t)$. If the probability is below a threshold, we ignore the cluster.
3) Region Identification: We then calculate the likelihood that the incident occurs in a hex cell $P(I, HC|T, D, W, M, C, t)$ (Eq. 7) using the learned conditional distributions of cells and the cluster-specific survival models.

$$P(I, HL|T, D, W, M, C, t) = P(HL|T, D, W, M, C, t, I) \times P(I|T, D, W, M, C, t) \tag{7}$$

Eq. 7 can be further simplified to Eq. 8 since we know which incident features affect the hex cells and incident probability.

$$P(I, HL|T, D, W, M, C, t) = P(HL|T, D, W, M, C) \times P(I|C, t) \tag{8}$$

Since we are predicting incidents at a future time, the weather at a future time may not be known precisely. In such cases, we predict the incident probabilities by summing over all possible weather conditions as given in Eq. 9.

$$P(I, HL|T, D, M, C, t) = \sum_{W} P(I, HL|T, D, W, M, C, t) \times P(W|T, D, M, C, t) \tag{9}$$

In Eq. 9, $P(W|T, D, M, C, t)$ represents the prior probability of the weather conditioned on Time, Day, Month, Cluster and prediction time. Since weather is independent of cluster type and prediction time, $P(W|T, D, M, C, t)$ can further be simplified to $P(W|T, D, M)$. The prior probability of weather can be obtained from historical weather data sets or any available weather prediction models. We used the same DarkSky database that provided incident weather data.

IV. DISCUSSION

The final prediction toolchain consists of two major components: the survival models for each cluster and the Bayesian network mapping cluster probabilities to the hex cells. We can validate each of these separately to show the correctness of the toolchain. We have already demonstrated the accuracy of the survival models: the likelihood analysis displayed in tables III-D and III-E shows that the survival models for each

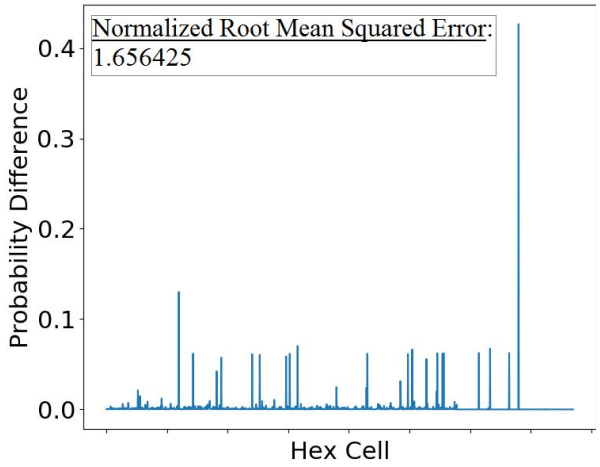

Fig. 6. Error in Predicted Hex Incident Probabilities vs. Validation Data

cluster match the incident data better than the popular negative binomial method [12]. This establishes that the survival models accurately represent the arrival times of accidents for each cluster.

To determine the accuracy of the Bayesian network analysis at predicting the distribution of incidents we compared its results to a validation set. This validation set consists of approximately 10 months of recent Nashville incident data, ranging from February 6th to December 23rd 2016. We ran the Bayesian analysis over this range of dates, and then compared its predicted accident distribution across hex locations against the actual distribution of the validation data. The results are presented in the figure 6. The bars represent the difference between the predicted probability and the actual probability for incidents in each hex cell. With a few exceptions, most cell's predicted incident probability is within 2% of the actual probability. There are a few cells that have slightly worse prediction performance, with the difference generally being less than 10%. These inaccuracies could be due to properties changing in the hex cells (increasing population density, for example), or may be due to having only two years of training data: future data should increase this accuracy. The normalized root mean squared error of the predicted distribution was **1.656425**, which shows that overall the predicted results match the validation data well.

A. Using the Toolchain

Now that we have demonstrated the accuracy of the toolchain components, we will discuss the toolchain's use. Since the toolchain is split into two major components, its use is also split into these two functions.

The first step for a user is inputting their current environmental factors (the current weather and time) and how much time they want to look into the future. The survival models are then consulted according to the analysis time. This determines the likelihood that an incident of each cluster will occur *at any location* within the given time. Then the Bayesian Network analysis is run using the input parameters to determine where an incident is likely to happen. By using these two components together, we can predict the likelihood of a next incident at a given time and location. Note that both of these components are necessary to use the toolchain. The survival likelihood for each incident of each cluster describes how likely each type of incident is, while the Bayesian analysis shows the probable distribution of any incident that might occur across the hex cells.

TABLE V
PREDICTION RAN WITH FOLLOWING PROPERTIES: WEATHER='CLEAR-DAY', DAY='THURSDAY', MONTH='MARCH', DATE='23RD', STARTTIME='15:00', ANALYSISTIME='2 HOURS'

Survival Models

Rank	Cluster	Incident Likelihood
1	1	0.6554
2	13	0.6294
3	7	0.49461
4	2	0.4448
5	6	0.2114

Hex Mapping

Rank	HexCell	Hex Probability
1	3523	0.05884
2	4140	0.05884
3	5491	0.04682
4	4703	0.04682
5	4699	0.04682

TABLE VI
PREDICTION RAN WITH FOLLOWING PROPERTIES: WEATHER='RAIN', DAY='THURSDAY', MONTH='MARCH', DATE='23RD', STARTTIME='15:00', ANALYSISTIME='2 HOURS'

Survival Models

Rank	Cluster	Incident Likelihood
1	1	0.6554
2	13	0.6294
3	7	0.49461
4	2	0.4448
5	6	0.2114

Hex Mapping

Rank	HexCell	Incident Probability
1	3513	0.13108
2	4332	0.13108
3	5290	0.13108
4	4803	0.13108
5	3587	0.13108

Example analysis from the trained toolchain: In table V we show the predicted accident distribution starting at 15:00 on March 23rd, on a Thursday, in clear weather, with an analysis time of 2 hours. The survival models indicate that there are a few clusters that have over a 50% likelihood of having an accident during this time segment. The Bayesian analysis then shows that cells 3523 and 4140 are tied for the most likely cells for an incident to take place in given the parameters, with cells 5491, 4703, and 4699 following close behind.

For comparison we provide the same analysis but during rainy weather in table VI. Because the analysis time is the same, the survival models for each cluster give the same accident probabilities. When determining which clusters are more likely, however, the Bayesian analysis considers the rainy weather. This changes the most likely cells to 3513, 4332, 5290, 4803, and 3587. These cells have a higher probability than in the clear weather case due to the increased likelihood of incidents during rain.

One last example demonstrated in table VII shows snowy

TABLE VII
PREDICTION RAN WITH FOLLOWING PROPERTIES: WEATHER='SNOW', DAY='THURSDAY', MONTH='JANUARY', DATE='10TH', STARTTIME='17:00', ANALYSISTIME='6 HOURS'

Survival Models

Rank	Cluster	Incident Likelihood
1	1	0.9591
2	13	0.9491
3	7	0.8710
4	2	0.8288
5	6	0.5100

Hex Mapping

Rank	HexCell	Incident Probability
1	4334	0.201918
2	3862	0.174190
3	4350	0.036615
4	3943	0.036615
5	3792	0.036615

weather in mid-January, given 6 hours of analysis time. Notice that due to the increased analysis time the survival models predict that incident's belonging to several clusters are very likely to happen. Looking at the cell distribution shows that there are two likely cells, but other cells are less likely than rainy or clear conditions. This might be caused by people using their cars less in snowy conditions (except in a few areas), although this is just speculation.

V. CONCLUSION

We have demonstrated that by combining clustering, survival analysis, and Bayesian network inference techniques a toolchain can be created that accurately forecasts incidents in both space and time. Unlike many popular techniques that focus on particular situations, the toolchain is shown to well over the spatially diverse Nashville metropolitan area. By leveraging this predictive model, in the future we will create more accurate dispatching algorithms to respond appropriately to motor vehicle accidents as they occur.

Acknowledgments: This work is sponsored by The National Science Foundation under the award number CNS-1640624. We express our gratitude and thanks for the support and advise of Prof. Yevgeniy Vorobeychik.

REFERENCES

[1] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annual Reviews in Control*, vol. 36, no. 2, pp. 220–234, 2012.

[2] K. L. Lane, "Identifying and supporting students at risk for emotional and behavioral disorders within multi-level models: Data driven approaches to conducting secondary interventions with an academic emphasis," *Education and Treatment of Children*, vol. 30, no. 4, pp. 135–164, 2007.

[3] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *Mobile Networks and Applications*, vol. 11, no. 3, pp. 327–339, 2006.

[4] R. Morris, M. F. Kaashoek, D. Karger, H. Balakrishnan, I. Stoica, D. Liben-Nowell, and F. Dabek, "Chord: A scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Transactions On Networking*, vol. 11, no. 1, pp. 17–32, 2003.

[5] D. Dhungana, G. Engelbrecht, J. X. Parreira, A. Schuster, R. Tobler, and D. Valerio, "Data-driven ecosystems in smart cities: A living example from seestadt aspern," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Dec 2016, pp. 82–87.

[6] T. H. Blackwell and J. S. Kaufman, "Response time effectiveness: comparison of response time and survival in an urban emergency medical services system," *Academic Emergency Medicine*, vol. 9, no. 4, pp. 288–295, 2002.

[7] S.-P. Miaou and H. Lum, "Modeling vehicle accidents and highway geometric design relationships," *Accident Analysis & Prevention*, vol. 25, no. 6, pp. 689–709, 1993.

[8] L.-Y. Chang, "Analysis of freeway accident frequencies: negative binomial regression versus artificial neural network," *Safety science*, vol. 43, no. 8, pp. 541–557, 2005.

[9] D.-G. Kim, Y. Lee, S. Washington, and K. Choi, "Modeling crash outcome probabilities at rural intersections: Application of hierarchical binomial logistic models," *Accident Analysis & Prevention*, vol. 39, no. 1, pp. 125–134, 2007.

[10] W. Ackaah and M. Salifu, "Crash prediction model for two-lane rural highways in the ashanti region of ghana," *IATSS research*, vol. 35, no. 1, pp. 34–40, 2011.

[11] Y. Qi, B. L. Smith, and J. Guo, "Freeway accident likelihood prediction using a panel data analysis approach," *Journal of transportation engineering*, vol. 133, no. 3, pp. 149–156, 2007.

[12] M. A. Abdel-Aty and A. E. Radwan, "Modeling traffic accident occurrence and involvement," *Accident Analysis & Prevention*, vol. 32, no. 5, pp. 633–642, 2000.

[13] H. C. Chin and M. A. Quddus, "Applying the random effect negative binomial model to examine traffic accident occurrence at signalized intersections," *Accident Analysis & Prevention*, vol. 35, no. 2, pp. 253–259, 2003.

[14] S. Raut and S. Karmore, "Review on: Severity estimation unit of automotive accident," in *Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in*. IEEE, 2015, pp. 523–526.

[15] L. Moreira-Matias and V. Cerqueira, "Cjammer-traffic jam cause prediction using boosted trees," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 743–748.

[16] Y. Chung, "Development of an accident duration prediction model on the korean freeway systems," *Accident Analysis & Prevention*, vol. 42, no. 1, pp. 282–289, 2010.

[17] A. Mukhopadhyay, Y. Vorobeychik, A. Dubey, and G. Biswas, "Prioritized allocation of emergency responders based on a continuous-time incident prediction model," in *Sixteenth International Conference on Antonomous Agents and Multiagent Sytems*, Sao Paulo - Brazil, 05/2017 2017.

[18] C. Li and G. Biswas, "Unsupervised learning with mixed numeric and nominal data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 4, pp. 673–690, 2002.

[19] R. O. Duda and P. E. Hart, *Pattern Elesslfication and Scene Analysis*. Wiley, 1973.

[20] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[21] D. Fisher and P. Langley, "Methods of conceptual clustering and their relation to numerical taxonomy." DTIC Document, Tech. Rep., 1985.

[22] D. W. Goodall, "A new similarity index based on probability," *Biometrics*, pp. 882–907, 1966.

[23] R. A. Fisher, *Statistical methods for research workers*. Genesis Publishing Pvt Ltd, 1925.

[24] H. Lancaster, "The combination of probabilities arising from data in discrete distributions," *Biometrika*, vol. 36, no. 3/4, pp. 370–382, 1949.

[25] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*. Springer, 2005.

[26] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[27] G. Shenyang, "Survival analysis (pocket guides to social work research methods)," 2010.

[28] D. Collett, "Modelling survival data," in *Modelling Survival Data in Medical Research*. Springer, 1994, pp. 53–106.

[29] L.-J. Wei, "The accelerated failure time model: a useful alternative to the cox regression model in survival analysis," *Statistics in medicine*, vol. 11, no. 14-15, pp. 1871–1879, 1992.

[30] W. Feller, *An introduction to probability theory and its applications*. John Wiley & Sons, 2008, vol. 2.

[31] Y. Wang and N. Nihan, "Quantitative analysis on angle-accident risk at signalized intersections," in *World Transport Research, Selected Proceedings of the 9th World Conference on Transport Research (in print)*, 2001.