Accident Analysis and Prevention 112 (2018) 39–49

Contents lists available at ScienceDirect

Accident Analysis and Prevention

journal homepage: www.elsevier.com/locate/aap

Real-time prediction and avoidance of secondary crashes under unexpected traffic congestion

Hyoshin Park[a,*], Ali Haghani[b], Siby Samuel[c], Michael A. Knodler[d]

[a] Department of Computational Science & Engineering, North Carolina Agricultural & Technical State University, United States
[b] Department of Civil & Environmental Engineering, University of Maryland, College Park, United States
[c] Department of Mechanical & Industrial Engineering, University of Massachusetts, Amherst, United States
[d] Department of Civil & Environmental Engineering, University of Massachusetts, Amherst, United States

ARTICLE INFO

Keywords:
Gradient boosting tree
Neural network
Variable importance
Rule extraction
Secondary incidents

ABSTRACT

According to the Federal Highway Administration, nonrecurring congestion contributes to nearly half of the overall congestion. Temporal disruptions impact the effective use of the complete roadway, due to speed reduction and rubbernecking resulting from primary incidents that in turn provoke secondary incidents. There is an additional reduction of discharge flow caused by secondary incident that significantly increases total delay. Therefore, it is important to sequentially predict the probability of secondary incidents and develop appropriate countermeasures to reduce the associated risk. Advanced computing techniques were used to easily understand and reliably predict secondary incident occurrences that have low sample mean and a small sample size. The likelihood of a secondary incident was sequentially predicted from the point of incident response to the eventual road clearance. The quality of predictions improved with the availability of additional information. The prediction performance of the principled Bayesian learning approach to neural networks (BNN) was compared to the Stochastic Gradient Boosted Decision Trees (GBDT). A pedagogical rule extraction approach, TREPAN, which extracts comprehensible rules from the neural networks, improved the ability to understand secondary incidents in a simplified manner. With an acceptable accuracy, GBDT is a useful tool that presents the relative importance of the predictor variables. Unexpected traffic congestion incurred by an incident is a dominant causative factor for the occurrence of secondary incidents at different stages of incident clearance. This symbolic description represents a series of decisions that may assist emergency operators by improving their decision-making capabilities. Analyzing causes and effects of traffic incidents helps traffic operators develop incident-specific strategic plans for prompt emergency response and clearance. Application of the model in connected vehicle environments will help drivers receive proactive corrective feedback before a crash. The proposed methodology can be used to alert drivers about potential highway conditions and may increase the drivers' awareness of potential events when no rerouting is possible, optimal or otherwise.

1. Introduction

1.1. Motivation

The integration of traffic and incidents databases provides an opportunity to examine the critical factors that cause incidents, and allows for the dynamic capture of traffic evolution as the primary incidents unfold. According to the Federal Highway Administration, nonrecurring congestion contributes to nearly half of the overall congestion. Temporal disruptions that take away part of the roadway from use are a major issue, because speed reduction and rubbernecking caused by primary incidents provoke secondary incidents. There is an additional reduction of discharge flow caused by secondary incident that significantly increases total delay. The likelihood of a secondary crash increases by 2.8% for each minute the primary incident continues to be a hazard (Khattak et al., 2012). To mitigate the impact of primary incidents in a timely manner, an optimal allocation of emergency response units (Park et al., 2016b) and performance measure (Park and Haghani, 2016b) have been proposed.

In addition to these post-crash strategy, it is important to prevent the secondary crash in advance with an effective prevention method. To enable this, we need to analyze the key cause with consideration of real-time traffic condition and incident duration (Fig. 1). There is a significant correlation between incident duration, the likelihood of a

* Corresponding author.
E-mail address: hpark1@ncat.edu (H. Park).

https://doi.org/10.1016/j.aap.2017.11.025
Received 13 May 2017; Received in revised form 15 October 2017; Accepted 18 November 2017
Available online 05 January 2018
0001-4575/

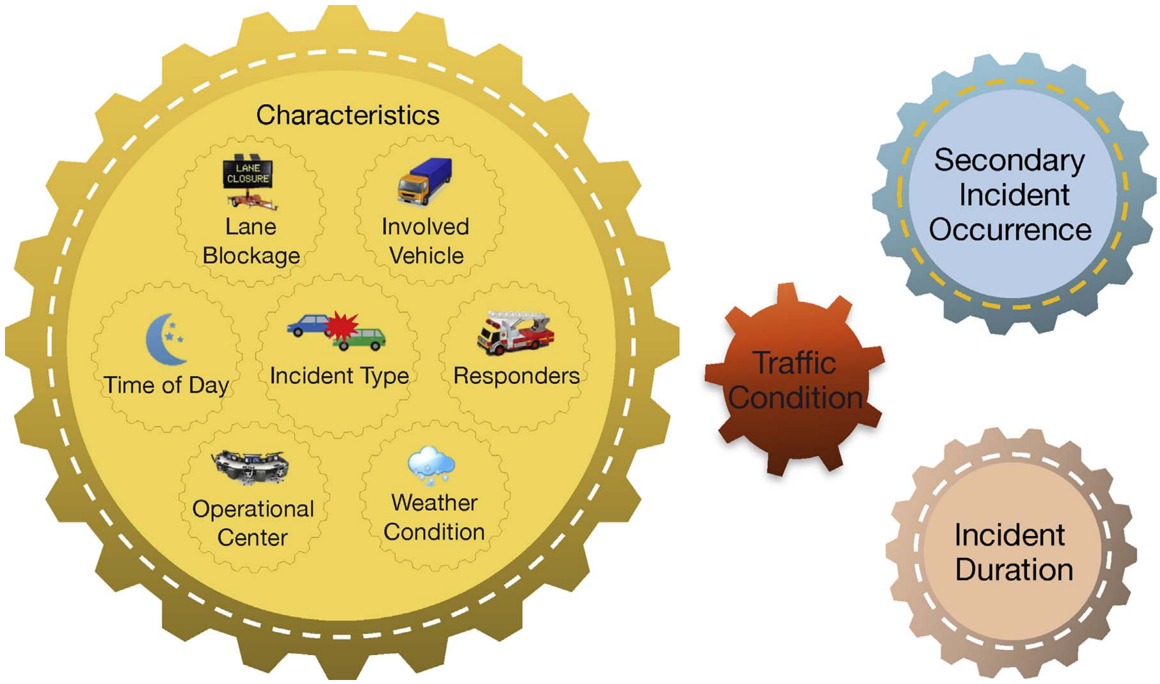

Fig. 1. Framework for studying the analysis and prevention of secondary crashes.

secondary incident, and the characteristics of a primary incident (Khattak et al., 2012). Various strategies (e.g., variable speed limits) have been proposed to reduce secondary crashes (Junhua et al., 2016; Sarker et al., 2017; Li et al., 2014). Accurate prediction and suggestions in easy to understand format helps traffic operators to develop incident-specific strategic plans for rapid emergency response and clearance.

Previously proposed prediction models for secondary incident likelihood have focused on the predictive power of the models and independent variables (Vlahogianni et al., 2012). Such singular prediction models assume that all information is available when the incident is reported. On the contrary, Park and Haghani (2016a) proposed a new model to continuously predict secondary incident likelihoods with the availability of additional information from the time point of incident response to the eventual road clearance.

Most solutions of machine learning algorithms have been perceived as black boxes. They might yield accurate predictions for the future, but the way those results are produced is hardly interpretable. A statistical model, in contrast, aims at quantifying the relation between the input and the expectation of the output via an interpretable function. The authors were faced with a critical issue of selecting the appropriate methodology and evaluating the efficiency of the developed model, thereby motivating the interpretation of machine learning models in statistical terms. In this study, the authors tested the accuracy and comprehensibility of models predicting secondary incident likelihoods. Accurate and understandable information provided by the tool may help emergency operators by improving their decision-making capabilities and lead to the development of collision warning systems that can prevent the occurrence of potential secondary events.

1.2. Background

Spatial and temporal influences of a primary incident on road users are closely related to occurrences of secondary incidents. Negative impacts of severe primary incidents cause congestion at upstream traffic, with a critical increase in secondary incident likelihoods. Since it takes time for upstream traffic to be congested, lower probability of secondary incidents with higher errors are expected at the initial incident clearance stages.

Defining and analyzing secondary incidents significantly depend on the performance of data collected from sensors. Previous studies commonly used inductive-loop detectors that are prone to various errors caused by malfunctions and communication failures. Recently, Park and Haghani (2016a) used real traffic data collected from vehicle probes for estimations of traffic congestion caused by incidents. The high-quality data collected from vehicle probe data, generally satisfying the requirements of applications for real-time travel time display, would yield reliable prediction results.

Previously, Vlahogianni et al. (2010), Imprialou et al. (2014), Yang et al. (2013a,b); Yang et al. (2014a,b), Junhua et al. (2016) focused on the classification of secondary incidents. While Dong et al. (2015) have studied safe driving behaviors within work zones, no predictions of secondary incident likelihood were made. To date, the research by Khattak et al. (2012), proposing linear models for the prediction of incident duration and secondary incident likelihood, continues to be among a handful of studies examining secondary incidence likelihood prediction. Compared to primary incidents, secondary incidents exhibit a lower sample mean and a small sample size. The wide variety of causes and impacts of nonrecurring congestion makes it difficult to quantify random and complex incident natures at a system level. As a result, crash prediction models have been over-fitted and have poor predictive performance. Since accident prediction models are non-normal and functional forms are typically nonlinear, it is shown that R^2 is not an appropriate measure (Miaou et al., 1996). It has been difficult to validate secondary incident occurrence and associated delays, owing to the lack of appropriate field data.

Recent studies have used neural network models to present top factors associated with secondary accident likelihood (Vlahogianni et al., 2012; Yang et al., 2014c). To avoid the over-fitting problem while retaining the interpretation of the model, two types of decision trees are used. While producing more understandable rules, decision trees discretize the classifier-separating hyperplane, thus leading to some information loss. In this study, two algorithms were used to overcome the shortcomings of traditional decision tree algorithms. First, Stochastic Gradient Boosted Decision Trees (GBDT), an ensemble method that combines the predictions of several models built with a given learning algorithm, was used to improve the generalizability over a single tree model. The importance of variables to prediction was estimated for different incident clearance stages.

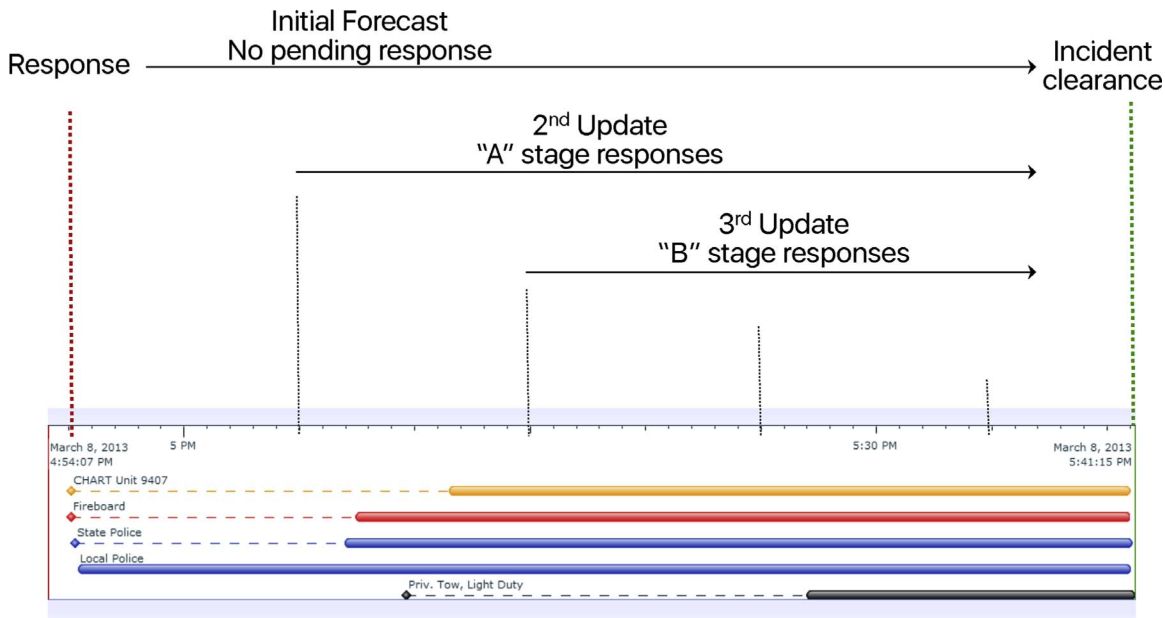

Fig. 2. Sequential forecasting framework.

Secondly, a principled Bayesian learning approach to neural networks (BNN) was used to improve the accuracy of the prediction models. Neural networks cannot be easily explained and have been regarded as black boxes. No satisfactory interpretation of neural networks behavior has been offered. A pedagogical rule extraction approach improves the understanding of the secondary incidents by extracting comprehensible rules from the neural networks. Compared to the GBDT, the proposed rule extraction algorithm branches the tree according to the predicted values by the neural network model and therefore retains a high level of accuracy while also allowing for an easier interpretation.

1.3. Sequential prediction framework

Predictions of secondary incident likelihoods depend on accurate clearance-duration characterization, and clearance duration depends on response-unit arrival time (Fig. 2).

Previous studies considered response time to be the time between when the responding agency was notified and when the first-response unit arrived at the scene. However, if the first responder is insufficient to clear the incident, clearance duration is extended until a second or greater responder arrives (Park et al., 2016a). In sequential prediction, each stage of prediction evaluates the response units present, and notified.

The authors also used INRIX data, which generally satisfies the requirements of applications for real-time travel time display (Park and Haghani, 2015). Using Bluetooth sensors as a ground truth, INRIX data showed that mean absolute value of the difference between the mean speed reported from the INRIX and the ground truth from Bluetooth sensors mean speed were within specification in all speed bins category (Haghani et al., 2013).

2. Methodology

The complex interactions among factors affecting prediction performance make modeling secondary incident likelihoods challenging. The authors introduce two advanced computing approaches.

2.1. Stochastic gradient boosted decision trees

Random forests, as an ensemble learning, generate a classification tree forest. Two well-known methods are bagging and boosting. In bagging, successive trees do not depend on earlier trees but are built independently using a bootstrapped sample of the data set. By contrast, in boosting methods, models are constructed sequentially, and one tries to reduce the bias of the combined model. The motivation is to combine several weak models to produce a powerful ensemble. The current effort uses stochastic gradient boosting decision trees (GBDT), which combine gradient boosting with bagging (Friedman, 2002). At each iteration, the base classifier is trained on a fraction subsample of the available training data.

Let $\{(x_1, y_1), \ldots, (x_i, y_i)\}_1^n$ be a set of incident data, consisting of output y_i (i.e., secondary crash occurrences) and input x_i (i.e., primary incident characteristics). Given a historical training sample, the authors' goal was to find a function $F(x)$ that minimizes the expected value of loss function $\Psi(y,\ F(x))$. Gradient tree boosting considers weak leaner, $h_m(x)$, for the function

$$F(x) = \sum_{m=1}^{M} \gamma_m h_m(x) \tag{1}$$

The authors then built the additive model in a forward stage-wise fashion

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{2}$$

At each stage, the decision tree $h_m(x)$ was chosen to minimize the loss function given the current model $F_{m-1}(x)$ and its fit $F_{m-1}(x_i)$

$$F_m(x) = F_{m-1}(x) + \text{argmin} \sum_{i=1}^{n} L(y_i,\ F_{m-1}(x_i) - h(x)) \tag{3}$$

At each iteration m, a tree partitions the x-space into L-disjoint regions and predicts a separate constant value in each one.

$$\gamma_{\text{im}} = \text{argmin}_{\gamma} \sum_{x_i \in R_{\text{im}}} \Psi(y_i,\ F_{m-1}(x_i) + \gamma) \tag{4}$$

Gradient boosting attempts to solve this minimization problem numerically via steepest descent. The steepest descent direction is the negative gradient of the loss function evaluated at the current model F_{m-1}, which can be calculated for any differentiable loss function. A shrinkage parameter υ was used to control the learning rate of the procedure. The stochastic gradient boosting incorporates randomness as an integral part of the procedure. A subsample of the training data was drawn at random from the full training data set. This randomly

Table 1
Gradient tree boosting algorithm.

GBDT	
Input:	Training data (x_i, y_i) for $i = 1, \ldots, n$, loss function $L(y, F(x))$, for total iteration of M
1.	initialization: $F_0(x) = \text{argmin}_\gamma \sum_{i=1}^{n} L(y_i, \gamma)$
2.	for $m = 1$ to M do, $y_{\text{im}} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \ldots, n$
3.	train $h_m(x)$ using training data (x_i, y_i) for $i = 1, \ldots, n$
4.	compute optimal solution of $\gamma_m = \text{argmin}_\gamma \sum_{i=1}^{n} L(y_i, F_{m-1(x_i)} + \gamma h_m(x_i)$
5.	update $F_m = F_{m-1(x_i)} + \gamma h_m(x_i))$
6.	endFor

selected subsample was then used, instead of the full sample, to fit the base learner and compute the model update for the current iteration. Table 1 shows algorithm starting with initialization of the model.

GBDT contains interpretable additive predictors. The partial effect of predictors is used to estimate the importance of each variable. To measure the importance of each variable after training, the values of the feature were permuted among the training data and the out-of-bag error was again computed on this perturbed data set. The importance score for the feature was computed by averaging the difference in out-of-bag error before and after the permutation over all trees. The score was normalized by the standard deviation of these differences. GBDT models were built in Python 3.3. The authors empirically found parameters to get the optimal values (learning rate: 0.01, maximum depth: 4, number of estimators: 170).

2.2. Neural networks decision trees

First, the authors introduce Bayesian neural networks for more accurate prediction. This study used a pedagogical rule extraction approach to improve understanding by extracting comprehensible rules from the neural networks.

2.2.1. Bayesian neural networks

Traditional neural networks are trained to obtain a near-optimal set of weights that minimize the error between the target values and network outputs. Back-propagation neural network (BPNN) models can fit the incident data with high levels of precision (Wei and Lee, 2007). Although BPNN has a good training result, it sometimes provides testing values with unacceptable variances (MATLAB). A principled Bayesian learning approach to neural networks (Park and Haghani, 2016a) is presented in this study to predict secondary incidents more accurately and robustly than traditional neural network models. The main difference between Bayesian neural networks (BNN) and BPNN is structural in nature, with the former exhibiting a variable structure and the latter employing a fixed structure. The multilayer perceptron, feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs was used.

Let $(x_1, y_1), \ldots, (x_n, y_n)$ be a set of incident data. Link function $f_B(x_i, \theta)$ can be obtained by following parameters: α_p, the parameter for the weights between the input layer, the bias, and the output layer, with normal prior distribution; β_j, the parameter for the weights between hidden layer and output layer; γ_{jh}, the parameter for the weights between the input layer, the bias, and the hidden layer; **P**, the input dimension; and **M**, the maximum number of hidden neurons specified by the user. x_{ik} is the kth element of the ith input. The link function is then represented as follows:

$$f_B(x_i, \theta) = \alpha_o + \sum_{i=1}^{p} \alpha_k \cdot x_{\text{ik}} + \sum_{i=1}^{m} \beta_j \cdot \tanh\left(\gamma_{\text{jo}} + \sum_{i=1}^{p} \gamma_{\text{jk}} \cdot x_{\text{ik}}\right) \tag{5}$$

Probabilistic learning models can be defined as a conditional distribution $P(y|x)$ for an output y, given the input vector x, and a standard deviation σ_i. $P(\theta|(x_1, y_1), \ldots, (x_n, y_n))$ is the posterior distribution of θ, given observed incident data (x_n, y_n). The posterior distribution for these parameters is proportional to the product of the prior distribution and the likelihood function, and it varies during training in response to how well a particular set of weights models the data. The predicted clearance time value $(\hat{y}_i)$ is given by:

$$\hat{y}_i = \int f_B(x_i, \theta) \cdot P(\theta|(x_1, y_1), \ldots, (x_n, y_n)) d\theta \tag{6}$$

When considering multilayer perceptrons, the ensuing complex posterior distributions θ make the evaluation of the above integrals difficult. More details of the prediction ability of neural networks can be found in Neal (2011).

2.2.2. Pedagogical interpretation of secondary incidents

Even if a trained model has learned interesting and possibly universal approximation properties, these relationships are encoded incomprehensibly as weight vectors and cannot easily support the generation of scientific theories. A decision tree is appealing when a good understanding of the process is essential, because it has self-explained properties rooted in the structure.

Neural network has an implicit decision of classification prediction and activation is used to represent the state of a unit at any given time. The activation moves forward from the input nodes, through the hidden nodes (if any) and to the output nodes. The activation of a hidden or an output unit is determined by passing its net input through a transfer function. In this study, local rule extraction method is used to decompose the neural network into a collection of networks, and extract rules presenting each of the constituent networks. Hidden and output units are approximated by threshold functions, and thus each unit can be described by a binary variable indicating whether it is secondary crash (activation = 1) or no secondary crash (activation = 0). TREPAN takes as input a model of the marginal probability density function $f(x_i)$ for each feature x_i along with the set of constraints that define allowable instances.

TREPAN treats the rule extraction from trained neural network as an inductive learning task to approximates the network. It uses oracle that can be used to answer queries during the learning process. Given a membership query, the oracle returns the class label for the instance (Craven and Shavlik, 1997). The target concept is the function represented by the network that serves as the oracle. Therefore, answering a membership query simply involves using the network to classify an instance. Membership queries are used in two different ways in TREPAN. Initially, they are used to get class labels for the networks training examples. Note that these class labels are not necessarily the true class labels, but instead they are determined by the networks classification of the instances. Since we are interested in inducing a description of the trained network, we treat the networks classifications as ground truth. TREPAN is not limited to using only the networks training data, however, it makes membership queries for other instances as well. TREPAN views the extraction task as one of learning. The probably approximately correct (PAC) model is used to provide an appropriate model for characterizing the scalability of TREPAN in terms of fidelity. With an efficient PAC for a rule-extraction algorithm, we can learn have a reasonable bound on how hard it is extract a representation of a given level of fidelity with a specified level of confidence. In this context, the rule-extraction algorithm is the learner, and the target function is the concept represented by the trained network. The goal in this task is to infer a representation of the target function that closely approximates it.

The notion of probably approximately correct learning assumed that

Table 2
TREPAN algorithm (Craven and Shavlik, 1997).

TREPAN	
Input:	Oracle(), training set S, feature set F, min sample, stopping criteria
1.	for each example $x \in S$
2.	class label for $x := \text{Oracle}(x)$
3.	initialize the root of the tree, R, as a leaf node
4.	construct a model M of the distribution of instances covered by node R
5.	query instances $R :=$ DrawSample({}, min sample $\|S\|$, M)
6.	use S and query instances R to determine class label for R
7.	initialize Queue with tuple (R, S, query instances R)
8.	**while** Queue not empty and global stopping criteria not satisfied
9.	remove node (N, S_N), query instances N, constraints N from Queue
10.	$T :=$ ConstructTest(F, $S_N \cup$ query instances N)
11.	make N an internal node with test T
12.	**for each** outcome, t, of test T
13.	make C, a new child node of N
14.	constraints $C :=$ constraints $N \cup \{T = t\}$
15.	$S_C :=$ members of S_N with outcome t on test T
16.	construct a model M of the distribution of instances covered by node C
17.	query instances $C :=$ DrawSample (constraints C, min sample $\|S_C\|$), M
18.	use S_C and query instances C to determine class label for C
19.	**if** local stopping criteria not satisfied **then**
20.	put (C, S_C, query instances C, constraints C) in Queue
	Return: tree with root R

there is an unknown target function f and an arbitrary probability distribution D over the instance space, and the goal of a learning algorithm is to infer a hypothesis h that closely approximates f. The learner is given access to examples of the target function through an oracle, $\mathbb{E}X(f, D)$, that randomly draws instances from distribution D and labels them using the target function f.

Previous incident duration studies have used decision trees to discover patterns in a given incident data set. Most of them are translated into If–Then–Else rules. TREPAN, a decision tree algorithm, accurately represents the network from which the rules are extracted (Table 2), becoming a useful tool for eliciting comprehensible representation of neural networks (Craven and Shavlik, 1997). The main difference between TREPAN and the classification and regression tree (CART) is that CART builds a tree from the original data, while TREPAN branches the tree according to the values predicted by the neural network model. Therefore, the decision tree retains good prediction performance of the actual neural networks. The extracted decision trees are comprehensible with high level of fidelity.

Instead of using the original training observations, TREPAN relabels training data according to the classifications made by the network. The relabeled data set is then used to initiate the tree-growing process. Training data become enriched with additional training instances, which are then also labeled by the neural network itself. The network is thus used as an oracle to answer class membership queries about artificially generated data points. Additional data from the oracle provides higher predictive accuracy, as the nodes lack sufficient data with the increase in tree size.

Traditional decision tree has a significant limitations by estimating marginal distribution globally In this paper, marginal distributions are estimated locally as it grows deeper into the tree. Instead of having an identical marginal distribution for all crash events, the tree stick to specific nodes by utilizing only the training examples that reach that node. In this way, we can capture the conditional dependency of that specific node. The marginal distribution of this new child nodes and ancestor nodes are compared using χ^2 test and Kolmogorov–Smirnov test (Craven and Shavlik, 1997). We only use local model when newly computed distribution is significantly different.

The process of expanding a node in TREPAN uses a best-first expansion, so that as it adds each node, it tries to maximize the gain in fidelity of the tree to the network that it is trying to model; a splitting test is selected for the node; and a child is created for each outcome of the test. Each child is either made a leaf of the tree or put into the queue for future expansion. Readers are referred to Craven and Shavlik (1997) for a more detailed description about expansion, splitting, stopping, and pruning of the algorithm.

3. Data

In the previous study (Park et al., 2016b), there was a statistical difference in clearance time when combinations of response units were delayed. A Tukey post-hoc test revealed that the time required to clear the incident was statistically significantly longer with particular delay types. This current study used the predicted duration of incident duration to predict the likelihood of a secondary incident. Sequential models updated predictions periodically, e.g., every 10 min. At each reevaluation point, the authors obtained the observed value. The quality of predictions improved as new information became available (e.g., response-unit arrival after travel time and damage to freeway infrastructure from traffic management center).

The authors obtained 1150 incidents as candidates for primary-secondary incident pairs. A total of 124 secondary incidents identified using the model proposed in Park and Haghani (2016a) were used in this study. Traffic message channel (TMC) sections provided the travel speed of vehicle operating as anonymous probes at 5-min intervals as a representation of the traffic state of each segment. According to INRIX quality indicator, only qualities satisfying a score of more than 20, a combination of real-time GPS and archival data are used in this study. Center for Advanced Transportation Technology Laboratory at the University of Maryland provided the traffic data from TMC codes and archived incidents. Prediction models had 13 variables. Contrary to previous studies that assumed a linear relationship between predictor variables and the likelihood of secondary incidents (Khattak et al., 2012), the prediction models used in this study were non parametric. A spatial distribution of primary and secondary incidents occurred between October 2012 and September 2013 along I-695 corridor is presented in Fig. 3. Along 51-mile section of the I-695 corridor, incidents occurring between exits are aggregated to closest exits.

Table 3 presents key contributing factors for sequential prediction.

Reduction of dimensions of the variables (e.g., factor analysis) may improve the accuracy of the model. However, this study focused on the original categories of the variable to interpret the result. Therefore, the above variables were used as is for training and testing both, the BNN and GBDT models.

Fig. 4 displays the presence of dependencies between incidents at different locations on freeway exits. The online dispatching algorithm utilized this dependency of incidents. The vertical line presents primary

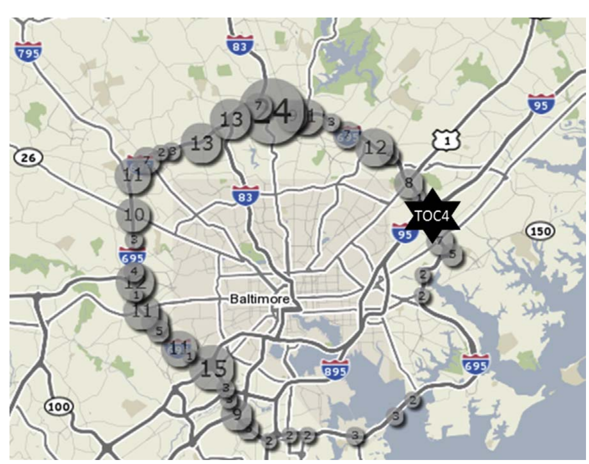

Fig. 3. Spatial distribution of Incidents on I-695 freeway.

Table 3
Key contributing variables to occurrence of secondary incidents.

Variables	Categories
Number of lanes blocked (BL)	No lane, 1 lane blocked, 2 lanes blocked …
Time of day (TOD)	Peak (1), day non-peak (2), night non-peak (3)
Traffic operation center (TOC)	TOC 4 (1), AOC (2), SOC (3)
Number of involved vehicles (NUM)	0, 1, 2, …
Truck involvement (TK)	No (1), one truck (2), more than one truck (3), truck overturn (4)
Location (AREA)	Exit 1–5 (1); Exit 6–10 (2); Exit 11–13 (3); Exit 14–18 (4); Exit 19–26 (5); Exit 27–31 (6); Exit 32–40 (7)
Incident (TYPE)	Collision (1), injury (2), fatality (3)
Require firefighter (FIRE)	Yes (1), no (2)
Severity (SEV)	Just off ramp closed (1), normal (2), guardrail damaged (3)
Traffic condition of the first upstream (FU)	Congested (1) not congested (2)
Traffic condition of the second upstream (SU)	Congested (1), not congested (2)
Incident was caused by traffic congestion (CTC)	No (1), yes (2)
Predicted clearance duration (CL)	0–5 (1), 5–10 (2), 10–20 (3), 20–30 (4), 30–40 (5) 40–50 (6), 50–60 (7), 60–70 (8), 70–80 (9), 80 plus (10)

EXIT	5	7	11	13	15	17	18	20	22	23	25	27	29	31	32	34	36
5	13.9	2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	6.5	1.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	4.6	3.3	2.7	2.0	2.0	2.7	2.0	1.3	0	0	0	1.3	0	1.3	0	0	0
13	0	0	5.2	17.1	6.5	2.6	1.3	1.3	0	0	0	0	0	1.3	0	0	0
15	0	2.6	3.9	5.2	9.1	5.2	2.6	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	10.8	4.6	1.5	0	0	0	0	0	0	0	0	0
18	0	2.9	1.0	0	0	3.8	9.5	4.8	0	0	1.9	0	0	1.9	0	0	0
20	0	2.9	0	0	0	0	1.9	1.9	2.9	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	1.8	3.6	5.4	0	0	0	0	0	0	0	0
23	1.0	2.1	0	0	2.1	1.0	0	3.1	7.2	0	0	0	2.1	1.0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	8.1	2.7	2.7	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	2.3	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	3.6	7.1	0	3.6	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.6	0	0
32	0	0	0	0	0	0	0	0	0	9.8	0	2.4	4.9	19.5	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	30.8	0	7.7	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.1	0

Fig. 4. The presence of dependencies between incidents at different locations on freeways: probability of secondary incidents (horizontal line) conditioned on primary incidents (vertical line) as a percentage.

incident locations, and the horizontal line presents the impact of primary incident locations on secondary incidents as a percentage. For example, when there was an incident at Exit 13, there was the potential for another incident at Exits 11–20. The total likelihood was 35.3% chance associated with this location.

The main contributing factors increasing likelihood of secondary incidents are rubbernecking factors and unexpected congestion from primary incidents. Primary incidents can cause secondary incidents in opposite direction traffic as well. In this paper, direction is not discretized. All incidents in the primary impact zone are defined as

Table 4
Evaluation of the performance of learning models.

Prediction model	Training		Testing	
	GBDT	BNN	GBDT	BNN
Average accuracy	85%	86%	84%	86%
Recall	0.93	0.96	0.93	0.96
Precision	0.94	0.93	0.95	0.92
F-value	0.94	0.95	0.94	0.94

secondary incidents including opposite direction (Yang et al., 2017).

4. Results

For identification of secondary incidents, the speed contour plot-based approach (Park and Haghani, 2016a) based on Gaussian Mixture Model and Bayesian structure equation model was also used to determine congestion patterns based on Inrix data. Among various identification models, speed contour plot-based approach was used to maximize the advantage of vehicle probe data by synthesizing the real-time traffic information and crash data to capture the prevailing traffic conditions. In this section the performance and interpretation of prediction models are presented.

4.1. Prediction performance

The MATLAB and modified NETLAB toolbox were used to implement BNN. The hybrid Monte Carlo method (HMC) returned 100 samples to form the posterior probability. For each run, the average computing time was 49 s (Core 2 Duo 3.00 GHz CPU and 8 GB memory). For optimal setting of the models, numbers were chosen after obtaining results from many tests that involved trying potential combinations of parameters. BNN employs hyperbolic tangent transfer function for the hidden units with 1 hidden layer, 11 hidden units, and logistic transfer function for the output units approximated by a threshold function for rule extraction. Instead of this arbitrarily threshold method, we consider the sign of the connection weight in the neural network to determine the influence of the variable on the activation of the unit. As presented in Craven and Shavlik (1997), the positive sign of weight implies that this input can only push the unit's activation toward 1, but not push it away from 1. In the same login, if the negative sign of weight implies that this input can only push the unit's activation away from 1, but not push it toward 1. For interpretation of an unit with an activation of 1, $\neg x_i$ is considered for inputs $\neg x_i$ that can push the activation of the output unit away from 1. In addition, non-negated x_i is considered for inputs that can push the activation of the output unit towards 1. Each iteration of the beam search considers $O(mn)$ modifications to the tests in the beam: there are n features, and in the more expensive case of a real-valued feature, there are $O(m)$ possible thresholds to consider on the feature.

Table 5
Performance of models for each update (GBDT and BNN).

Clearance	GBDT			BNN		
	True	False		True	False	
		Primary	Secondary		Primary	Secondary
0–5 min	82.10%	13.10%	4.80%	81.60%	13.00%	5.40%
5–10 min	83.20%	11.20%	5.60%	82.90%	11.10%	6.00%
10–20 min	84.70%	8.50%	6.80%	84.90%	8.50%	6.60%
20–30 min	84.10%	7.20%	8.70%	84.50%	7.30%	8.20%
30–40 min	84.20%	8.20%	7.60%	84.50%	8.30%	7.30%
40–50 min	89.80%	4.20%	6.00%	92.10%	4.30%	3.60%
50–60 min	90.10%	5.80%	4.10%	90.30%	5.80%	3.90%
60–70 min	88.40%	6.70%	4.90%	88.50%	6.70%	4.80%
70–80 min	84.90%	10.40%	4.70%	85.10%	10.40%	4.50%
80 min +	80.50%	16.60%	2.90%	81.00%	16.70%	2.40%

It is well known that each running multiple neural networks may produce different results. Thus, neural network models were run 10 independent times to get average performance. After testing various learning rates, 0.1 was selected for GBDT. To evaluate the temporal transferability of the models, after trial and error of training and testing different sizes of data, the authors randomly split the data into two parts: 70% of the data set for training and 30% for testing set.

One-time and sequential-prediction models were investigated using the nine variables, except for response delay type, in the clearance time prediction. In addition, the following four variables were added. Traffic condition variables were found to influence significantly, the probability of having a secondary incident (Vlahogianni et al., 2012). In contrast to the models used in Vlahogianni et al. (2012), this study used sequentially predicted clearance duration to predict the probability of having a secondary incident.

Two models were trained on the data set that had a perfect information of a detected incident and actual traffic condition of upstream when a secondary incident occurred. On the exclusive data, each

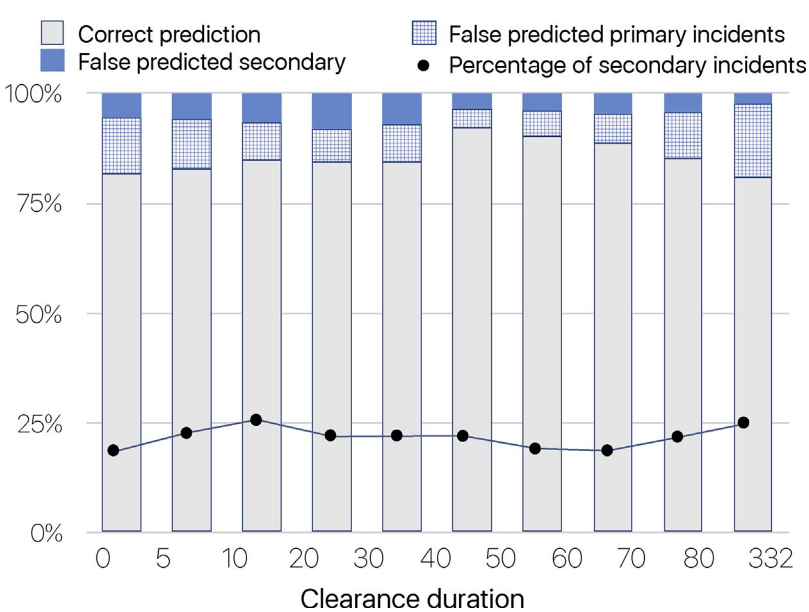

Fig. 5. MAPE performance of models with different stages of clearance duration.

```
If Occ Diff<0.5)},
    If  Lane Blockage = No lane, 1 lane
        If  Type=CPD, Disabled
                ⋮
        Else if  Type=CF, CPI, Fire, Other
                ⋮
    Else if  Lane Blockage = 2 lanes, 3 lanes, 4 lanes
        If  Type=CPD, Disabled
                ⋮
        Else if  Type=CF, CPI, Fire, Other
                ⋮
Else if Occ Diff ≥ 0.5)},
    If  Lane Blockage = No lane, 1 lane
        If  Type=CPD, Disabled
                ⋮
        Else if  Type=CF, CPI, Fire, Other
                ⋮
    Else if  Lane Blockage = 2 lanes, 3 lanes, 4 lanes
        If  Type=CPD, Disabled
                ⋮
        Else if  Type=CF, CPI, Fire, Other
                ⋮
```

Fig. 6. Extracted If–Then–Else rules for second split from decision tree.

model was tested for predictive power. The BNN model outperformed GBDT in both training and testing (Table 4). The result may be due to no explicit regularization in the GBDT algorithm. Implicit regularization may not be effective, because the procedure separated tree learning and forest learning (Friedman, 2002). GBDT has shown relatively poor performance on high-dimensional sparse data.

In the previous study Park and Haghani (2016a), only the average accuracy of BNN was analyzed. This study additionally tested other performance measures: *F*-measure as the harmonic mean of the probability of correctly labeling the detection (recall) and probability that a positive prediction is correct (precision) (Table 4). The model performance was approximately 85% accurate on average. The fairly balanced trade-off between precision and recall shows the robustness of the model.

The authors sequentially tested the performance (true positive) using trained BNN (Fig. 5). Once an emergency response unit arrives at the incident scene, the clearance starts. The proportion of false predicted primary incidents continuously decreased as new information (e.g., traffic condition upstream) updated, until the clearance stage became more than 60 min. However, after 60-min clearance duration, both errors increase resulting in prediction performance as low as the clearance duration less than 5 min. The increase in error stems from a relatively smaller sample size of secondary incidents after 60 min. This is caused by lack of information quantity and accuracy. In earlier stage, more incidents are falsely predicted to be primary incidents. On the contrary, in later stage with clearance duration of 40 min, more incidents are falsely predicted to be secondary incidents. After a primary incident occur, there is a certain amount of time may passes for secondary incident the most likely occur (Ng et al., 2013). As the clearance duration is longer, capacity reduction due to lane block cause more vehicles to remain in a queue, and more congestion increases the likelihood of secondary incidents. From empirical distribution of our data, secondary incidents are more likely to occur when the clearance duration is between 10 and 20 min, or more than 75 min.

The authors tested trained parameters on different stages of incident clearance. From the results in Table 5, it is apparent that the prediction performance continuously improved as new information (e.g., traffic condition upstream) updated, until the clearance stage became more than 60 min. BNN outperformed GBDT, except for the first two clearance stages (i.e., 10 min), after the primary incident occurrence. BNN tended to underestimate when there was no congestion upstream of the incident scene caused by the negative impact of the primary incident.

4.2. Interpretation of prediction models

Pedagogical interpretation is one of the most powerful interpretation tools in a decision tree. The authors present a comprehensive summary of TREPAN and GBDT.

4.2.1. Interpretation of decision trees: rule extraction

TREPAN parameters were set as follows: at least 200 instances (training examples plus queries) were considered before selecting each split; significance level for comparing m-of-n tests was set to 0.05; maximum tree size was set to 35 internal nodes, which is the size of a complete binary tree of depth six. The extracted tree showed high fidelity (90.9%) to the network from which it was generated, resulting in 25 total nodes and 13 leaves.

Assuming three main contributors for secondary incidents: location, time of day, and type a decision tree can be built using If–Then–Else, commonly used in CART (Fig. 6).

As presented in Park and Haghani (2016a), the "M-of-N" rule has a better efficiency than traditional decision tree rules. "If–Then–Else" statements in Fig. 6 were transformed to extracted decision trees presented in the M-of-N rule (e.g., Boolean features: location, time of day, and type). Two of {(Location (area 3) = Exit 11, or 12, or 13), (Time of Day = peak hour), and (Type (1) = Collision with property damage)}

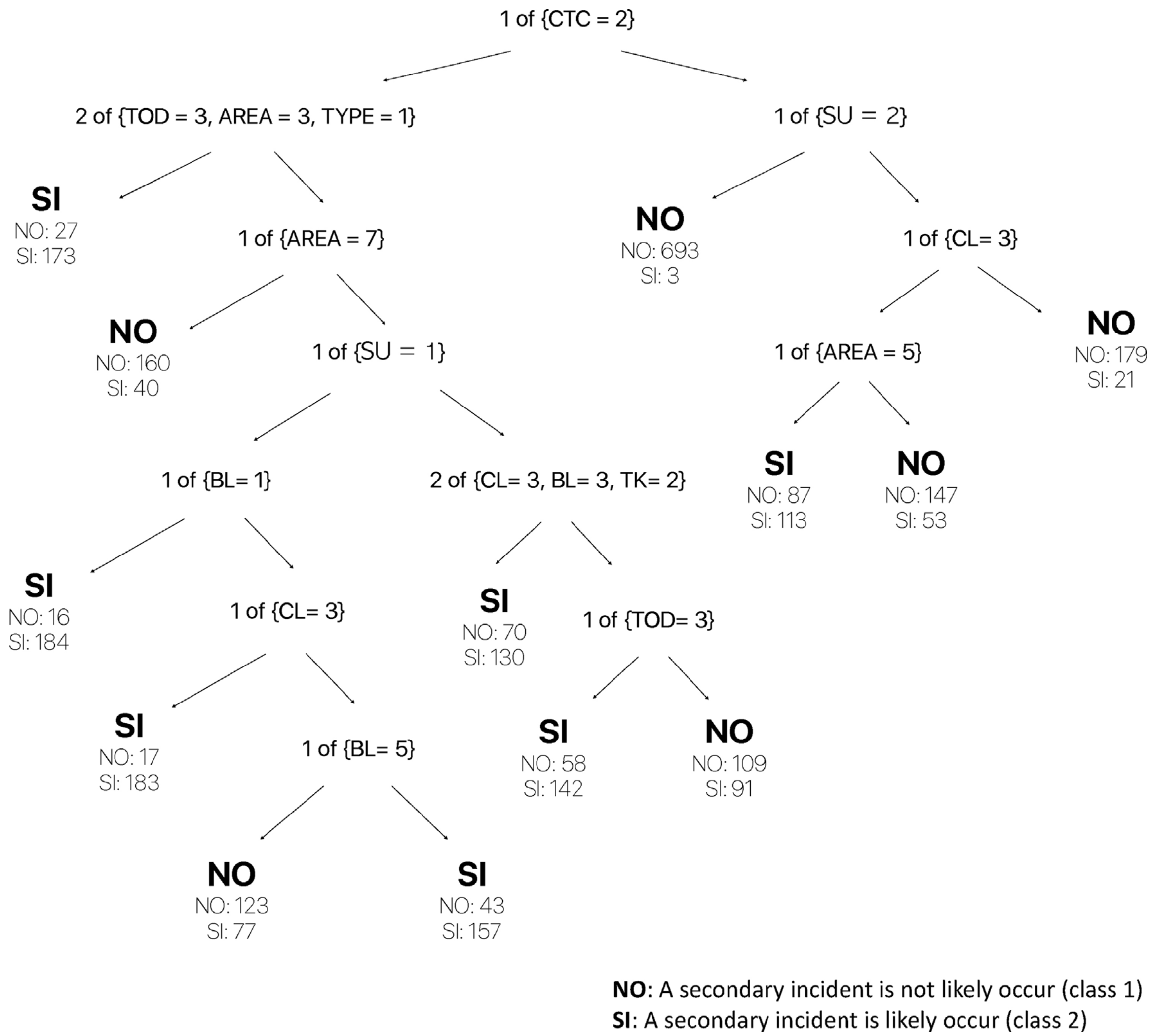

Fig. 7. Extracted decision tree from prediction (Park and Haghani, 2016a).

are logically equivalent to: "{(Location = Exit 11, or 12, or 13) and (Time of Day = peak hour)}", or "{(Location = Exit 11, or 12, or 13) and (Type = Collision with property damage)}", or "{(Time of Day = peak hour) and (Type = Collision with property damage)}".

If this condition is satisfied, one reaches the leaf node, which is classified to class 2 (secondary incident). Fig. 7 presents that the occurrence of secondary incidents (SI) was predicted to be 70.5% (173 among total 200 incidents). Each node was assigned a priority, defined to be the proportion of examples misclassified by the node (Craven and Shavlik, 1997). To decide how to partition the part of the instance space by the internal node, the M-of-N search used information gain as its heuristic evaluation function. The search process selected the best binary test at the current node to maximize the information gain. Whenever the number inside of the bracket was not sufficient to draw the sample distribution, the model got extra sampled distribution from membership queries (outside of the bracket).

In general, the GBDT is composed of many single decision trees (e.g., 170 CART as estimators in this study). Unlike a single CART, the GBDT can capture nonlinear interaction between the predictors and secondary incident occurrences. As we have more estimators, we can obtain higher accuracy, but training and prediction time also grows. In this study, there was no improvement with more than 170 estimators. Most of the running time were within 10 sec that can be sequentially applied in real-time.

It is clear that GBDT with "If–Then–Else" statements has more decision points; as a result, the M-of-N expressions better facilitate comprehensibility of the tree. In this way, TREPAN reduces the tree depth compared to the "If–Then Rules" statements used in the CART and GBDT. For sparse data with more than hundreds of categories, GBDT may need a pre-processing on input predictors. TREPAN is better in dealing with variables with high dimension. Thus, TREPAN rules are straightforward to code in any incident management software. Traffic operators can easily understand TREPAN outputs by following the branches related to the conditions of variables. Moreover, this tool can also generate predictions when only partial information is available, since each node generates the maximum likelihood estimation of how long the incident may last. This information may contribute to the accurate selection of appropriate emergency response units.

4.3. Interpretation of GBDT: the relative importance of critical factors

GBDT has used another pedagogical interpretation tool to measure the importance of variables. A comprehensive summary of GBDT's dependence on the joint values of the input variables is presented (Fig. 8).

Regardless of the clearance stages, main effects that explain the secondary incident occurrences were from the decision on whether or not the primary incident mainly caused the congestion on the road. However, the relative contribution of this predictor variable, the main cause of congestion, became less significant within the group of shorter clearance stages (i.e., more than 5 min). Instead, the relative contribution of the predictor variable, the traffic condition of the first or second exit upstream from the incident location, became more

Fig. 8. Relative importance.

significant.

The decision tree is a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by logic. By contrast, GBDT treats the decision tree model as a black box. It is hard to interpret, and it does not take advantage of the tree structure itself. Use of small shrinkage parameter GBDT could lead to a huge tree model, which is very undesirable, as it leads to high computational cost of applications.

Future study includes removing relatively less important variables to make the process simpler.

5. Practical applications

The decision rules are cast in a form that appears to be particularly suitable for the representation of an incident that requires quick and concise action. Since each incident is different, the sequence of individual responder actions depends upon a variety of factors, such as who arrives first on scene, the severity of the incident, and the surrounding traffic conditions, among others.

Drivers can receive proactive corrective feedback before a crash. A user can simply insert the values for different parameters into a tree and obtain the results (Fig. 9). The on-board unit in a connected vehicle environment or the smartphone application (e.g., WAZE) can help drivers navigate around road closures and get where they need to be. If the likelihood of secondary incidents is high, notifications like “watch out”, or “pay attention”, could make driving safer. Moreover, a connection weight approach accurately quantifies the contributions each variable makes from the neural network. On the other hand, the drivers can respond to the user interface, and this ground truth data can be used for improving the accuracy of the secondary incident prediction.

6. Conclusion and future research directions

The authors highlight the concept of decision tree learning, which has several advantages, including being able to handle heterogeneous data with ease when different features are assimilated from different sources. Disadvantages of traditional decision tree learning, mainly the difficulties associated with achieving the most accurate prediction performance, were improved in this study. The authors applied stochastic gradient boosting and rule extraction techniques. Developed models provided the authors with accurate and comprehensible predictions of secondary incident occurrences. BNN has accurate prediction results and extracted pedagogical rules provide a series of decisions to assist emergency operators and improve their decision-making capabilities. Unexpected traffic congestion incurred by an incident is a dominant causative factor for the occurrence of secondary incidents at different stages of incident clearance. The relative importance of predictor variables extracted from GBDT may help decide which incident cases have more priority under resource limitations, and highlight potential factors to be improved. On a real-time basis, the secondary incident likelihood is sequentially predicted as new parameters are updated. The update of real-time information reveals clearance progress, queue formation, and traffic recovery. Revealing the cost of secondary incidents is essential for convincing emergency planners to take into account secondary incidents when they design incident management systems.

The proposed methodology can be used to alert drivers about potential highway conditions and may increase the drivers’ awareness of potential events when no re-routing is possible, optimal or otherwise. The potential future studies are as follows.

- The developed model can be extended to analyze the relative significance of the different input variables to the performance of the secondary incident prediction model (Park and Haghani, 2016a; Vlahogianni et al., 2012). Based on dependences on each location, secondary incidents can be considered in emergency response policy (Park et al., 2016b). Incorporating high-resolution CCTV images will allow more accurate updates of incident response and clearance process.
- Future research may also examine the utility of such approaches in studying the operational and safety aspects of extended automated and connected roadway systems. There exists the potential to train vehicle automation to make decisions that minimize secondary incidents and therefore optimize travel time.
- Advanced data storage will enable us to archive many years of incident data as well as traffic data. With state-wide large scale network, more critical factors can be investigated to find whether the likelihood of secondary incident in the opposite direction is different from same direction of the primary incident.
- Emerging technologies have provided public more critical variables influencing crashes, and as a result complexity of model have

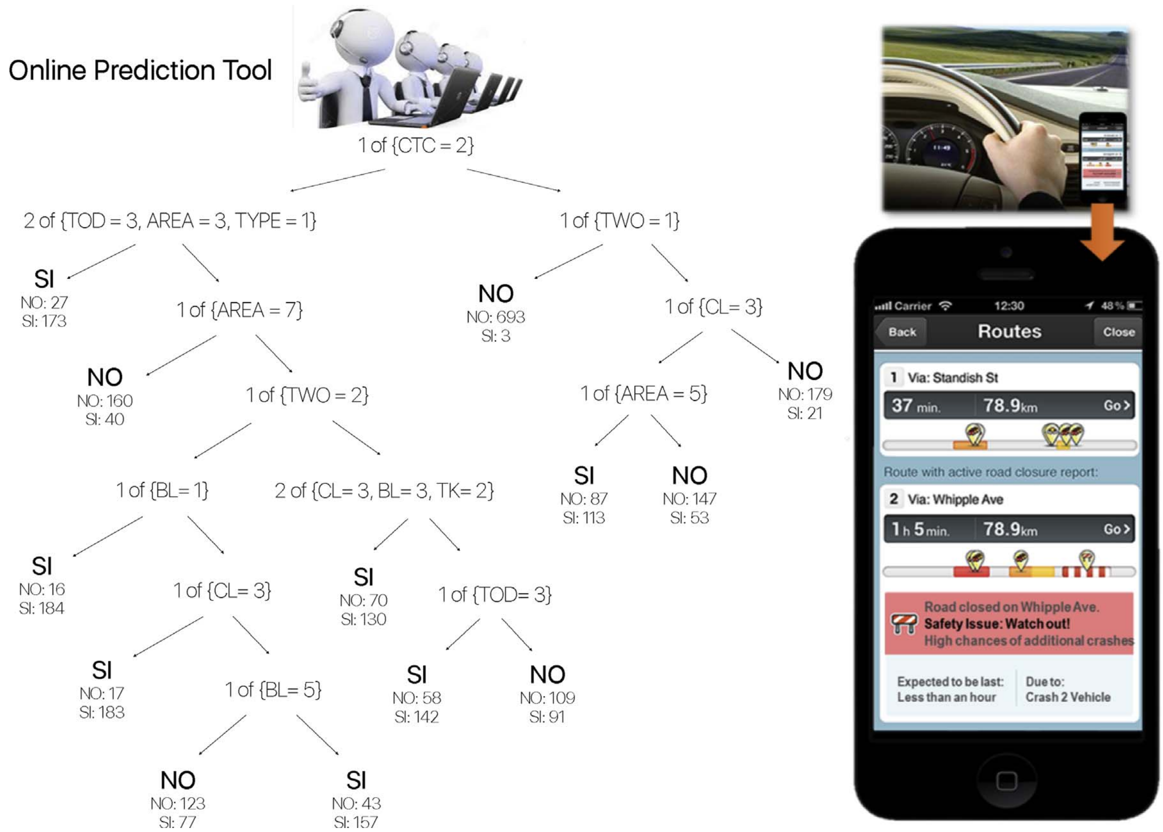

Fig. 9. Application of incident online prediction tool.

increased. To deal with complexity in a larger scale network model with potential over-fitting problem, principle component analysis or introducing additional information (regularization) can be tested to see how much of problems can be remedied.

- Highway capacity manual models are deterministic and cannot be applied in real time because input parameters must be known. Based on the capacity reduction proposed by Park and Haghani (2016b), future research can conduct simulation to consider spatial characteristics and estimate total delay impact sorely caused by secondary incidents.
- Additionally, the effectiveness of real-time interventions delivered via smartphone applications can be evaluated to understand how to reduce the crash likelihood associated with secondary incidents, not just those stemming from recurring congestion, but also the incidents resulting from nonrecurring congestion.

References

Craven, M., Shavlik, J., 1997. Understanding time-series networks: a case study in rule extraction. Int. J. Neural Syst. 4, 373–384.

Dong, S., Wang, H., Hurwitz, D., Zhang, G., Shi, J., 2015. Nonparametric modeling of vehicle-type-specific headway distribution in freeway work zones. J. Transp. Eng. 141 (11), 05015004.

Friedman, J., 2002. Stochastic gradient boosting? Comput. Stat. Data Anal. Nonlinear Methods Data Min. 38 (4), 367–378.

Haghani, A., Hamedi, M., Park, H., Aliari, Y., Zhang, X., 2013. I-95 Corridor Coalition Vehicle Probe Project: Validation of INRIX Data Monthly Report: Georgia. https://doi.org/10.13140/RG.2.2.11544.70400.

Imprialou, M., Orfanou, F., Vlahogianni, E., Karlaftis, M., 2014. Methods for defining spatiotemporal influence areas and secondary incident detection in freeways. J. Transp. Eng.

Junhua, W., Boya, L., Lanfang, Z., Ragland, D.R., 2016. Modeling secondary accidents identified by traffic shock waves. Accid. Anal. Prev. 87, 141–147.

Khattak, A.J., Wang, X., Zhang, H., 2012. Incident management integration tool: Dynamically predicting incident durations, secondary incident occurrence and incident delays. IET Intell. Transp. Syst. 6 (2), 204–214.

Li, Z., Li, Y., Liu, P., Wang, W., Xu, C., 2014. Development of a variable speed limit strategy to reduce secondary collision risks during inclement weathers. Accid. Anal. Prev. 72, 134–145.

Miaou, S., Lu, A., Lum, H., 1996. Pitfalls of using R2 to evaluate goodness of fit of accident prediction models. Transp. Res. Rec. J. Transp. Res. Board 1542, 6–13.

Neal, R.M., 2011. MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo. Chapman and Hall, CRC Press.

Ng, M.W., Khattak, A.J., Talley, W.K., 2013. Modeling the time to the next primary and secondary incident: a semi-Markov stochastic process approach. Transp. Res. B: Methodol. 58, 44–57.

Park, H., Haghani, A., 2015. Optimal number and location of bluetooth sensors considering stochastic travel time prediction. Transp. Res. C: Emerg. Technol. 55, 203–216.

Park, H., Haghani, A., 2016a. Real-time prediction of secondary incident occurrences using vehicle probe data. Transp. Res. C: Emerg. Technol. 70, 69–85.

Park, H., Haghani, A., 2016b. Stochastic capacity adjustment considering secondary incidents? IEEE Trans. Intell. Transp. Syst. 17 (10), 2843–2853.

Park, H., Haghani, A., Zhang, X., 2016a. Interpretation of bayesian neural networks for predicting the duration of detected incidents? J. Intell. Transp. Syst. 20 (4), 385–400.

Park, H., Shafahi, A., Haghani, A., 2016b. A stochastic emergency response location model considering secondary incidents on freeways? IEEE Trans. Intell. Transp. Syst. 17 (9), 2528–2540.

Sarker, A.A., Paleti, R., Mishra, S., Golias, M.M., Freeze, P.B., 2017. Prediction of secondary crash frequency on highway networks. Accid. Anal. Prev. 98, 108–117.

Vlahogianni, E., Karlaftis, M., Golias, J., Halkias, B., 2010. Freeway operations, spatiotemporal-incident characteristics, and secondary-crash occurrence. Transp. Res. Rec. J. Transp. Res. Board 2178, 1–9.

Vlahogianni, E.I., Karlaftis, M.G., Orfanou, F.P., 2012. Modeling the effects of weather and traffic on the risk of secondary incidents? J. Intell. Transp. Syst. 16 (3), 109–117.

Wei, C., Lee, Y., 2007. Sequential forecast of incident duration using artificial neural network models? Accid. Anal. Prev. 39 (5), 944–954.

Yang, H., Bartin, B., Ozbay, K., 2013a. Investigating the characteristics of secondary crashes on freeways. Transportation Research Board 92nd Annual Meeting. Compendium of Papers CD-ROM.

Yang, H., Bartin, B., Ozbay, K., 2013b. Use of sensor data to identify secondary crashes on freeways. Transp. Res. Rec. J. Transp. Res. Board 2396, 82–92.

Yang, H., Bartin, B., Ozbay, K., 2014a. Mining the characteristics of secondary crashes on highways. J. Transp. Eng. 140 (4), 04013024.

Yang, H., Ozbay, K., Morgul, E., Bartin, B., Xie, K., 2014b. Development of online scalable approach for identifying secondary crashes. Transp. Res. Rec. J. Transp. Res. Board 2470, 24–33.

Yang, H., Ozbay, K., Xie, K., 2014c. Assessing the risk of secondary crashes on highways Proceedings of the International Conference on Road Safety (RSS2013). J. Saf. Res. 49 143.e1–149.e1.

Yang, H., Wang, Z., Xie, K., Dai, D., 2017. Use of ubiquitous probe vehicle data for identifying secondary crashes. Transp. Res. C: Emerg. Technol. 82, 138–160.