Data Mining and Machine Learning Techniques

Spring 2022

Xiao Qin, Ph.D., P.E.
Professor, University of Wisconsin-Milwaukee

How Tesla Autopilot Sees En-route!

Tesla uses PyTorch* for distributed CNN training. A full build of Autopilot NN involves 48 networks, takes 70,000 GPU hours to train, and produces 1,000 distinct tensors (predictions) at each timestep.

(Source: FSD Beta 10.9 drive by @WholeMarsBlog)

*PyTorch is a free and open-source machine learning framework based on the Torch library. Primarily developed by Facebook's AI Research lab, it is used for applications such as computer vision and natural language processing. (source: Wikipedia)

Learning Objectives

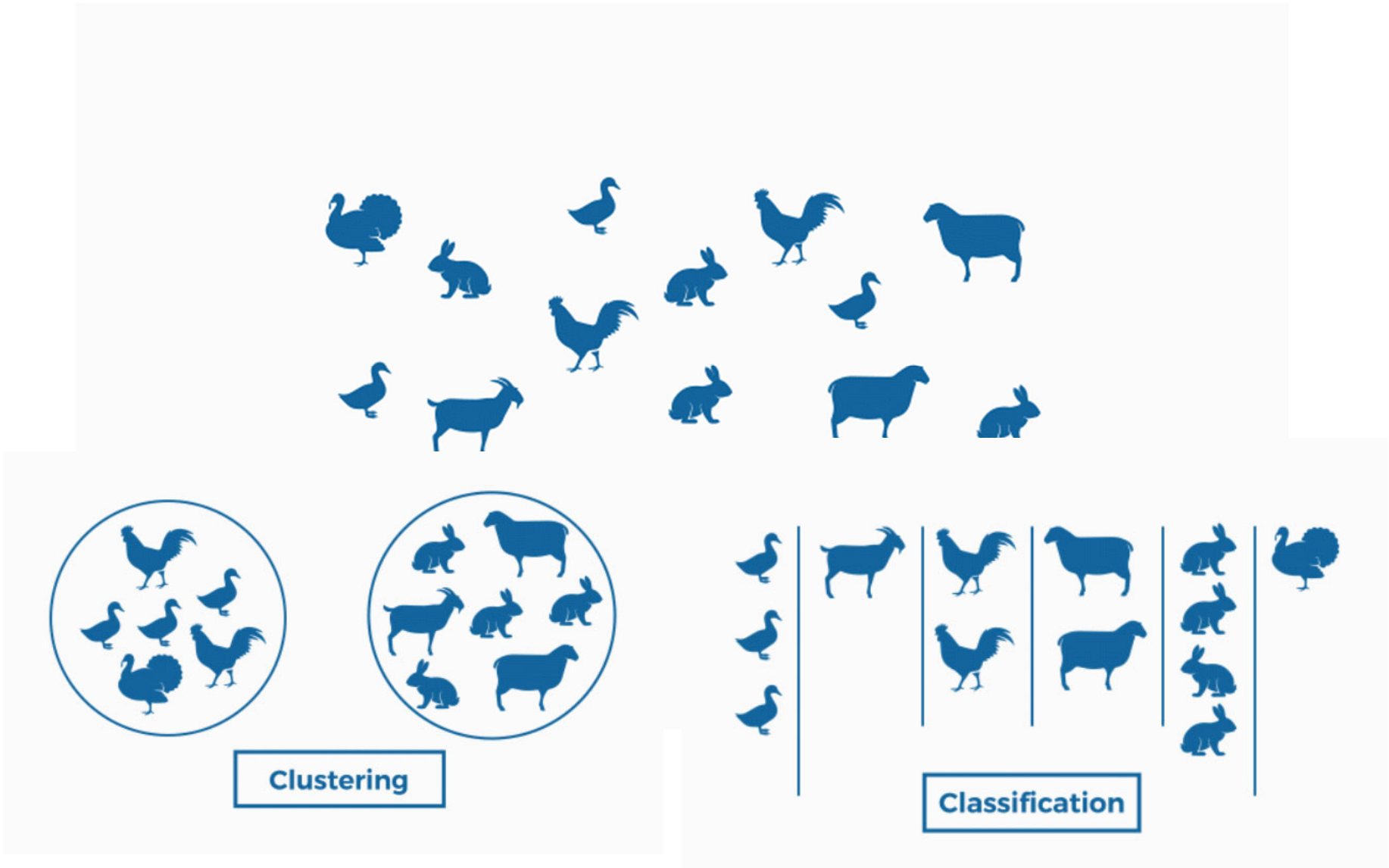
- Be familiar with machine learning methodologies and techniques that have been used to analyze crash data, including *association rules, clustering analysis, decision tree, Bayesian networks, neural networks, and support vector machines*.
- Learn to solve your problem based on known theories while keeping practical considerations in mind.
- Select appropriate methods and techniques.
- Know how to implement in statistical software *R*.

Introduction

- Data mining is the process of discovering and extracting useful information from a vast amount of data.
- Machine learning incorporates the principles and techniques of data mining and uses the same algorithms to automatically learn from and adapt to the data.
- The characteristics in machine learning include, but are not limited to:

Introduction (cont'd)

- *Clustering*: create clusters where similar data points are grouped together. Applications include crash pattern recognition and hot spot identification. Typical methods include k-means and latent class clustering.
- *Classification*: assign data to one of predefined categories. Applications include real-time crash prediction, injury severity prediction, crash types, and risky driver behaviors. Typical methods include logistic regression, support vector machines, neural networks, random forests, and Bayesian network classifier.
- *Regression*: predict continuous quantities from input data. Applications include prediction of crash counts, rates and certain types of crash events. Typical methods include generalized linear regression, neural networks, and Gaussian processes.



Source: Classification Vs. Clustering - A Practical Explanation
https://blog.bismart.com/en/classification-vs.-clustering-a-practical-explanation#:~:text=Although%20both%20techniques%20have%20certain,which%20differentiate%20them%20from%20other

Association Rules

- A rule-based machine learning method for discovering relations between variables in large databases.
- Specifically, it identifies the relative frequency of sets of variables (e.g., highway geometric features, traffic conditions, driver characteristics) occurring alone and together given an event such as a traffic crash.

Association Rules

Rules have the form A → B where A is antecedent, and B is consequent.

$$Support\ (A \rightarrow B) = \frac{\#(A \cap B)}{N}\ ;\ Confidence = \frac{Support\ (A \rightarrow B)}{Support\ (A)}\ ;\ Lift = \frac{Support\ (A \rightarrow B)}{Support\ (A) \times Support\ (B)}$$

where N is the number of crashes and $\#\ (A \cap B)$ is the number of crashes in which both Conditions A (antecedent) and B (consequent) are presented.

For example, in the rule "alcohol → reckless driving"
@ uncontrolled intersection: support = 3%, confidence = 51%, lift = 1.5,
@ sign-controlled intersection support = 2%, confidence = 80%, lift = 1.2
@ signal-controlled intersection: support = 2%, confidence = 48%, lift = 1.3

- Lift indicates that reckless driving is positively associated with alcohol at all intersections regardless of traffic control type.
- Uncontrolled intersections have the highest support value (i.e., the proportion of crashes including both reckless driving and alcohol is 3% in all crashes); while sign-controlled intersections have the highest confidence value (i.e., among all DUI crashes, reckless driving accounts for 80%)

Bayesian Networks (BN)

- A probabilistic graphical model that depicts a set of variables and their conditional dependencies via a *directed acyclic graph (DAG)*.
- BN has two main components:
  - the causal network model (topology)
  - the conditional probability tables (CPTs).
- The model causal relationships are represented as DAGs in which variables are denoted by nodes, and relationships (e.g., causality, relevance) among variables are described by arcs between nodes.
- CPTs explicitly specify the dependencies among variables in terms of conditional probability distributions.
- The R package includes “bnlearn” to help explain the graphical structure of BN, estimate its parameters and perform useful inferences. Functions in “bnlearn” include HC and tabu, as well as severity score functions such as AIC and BIC.

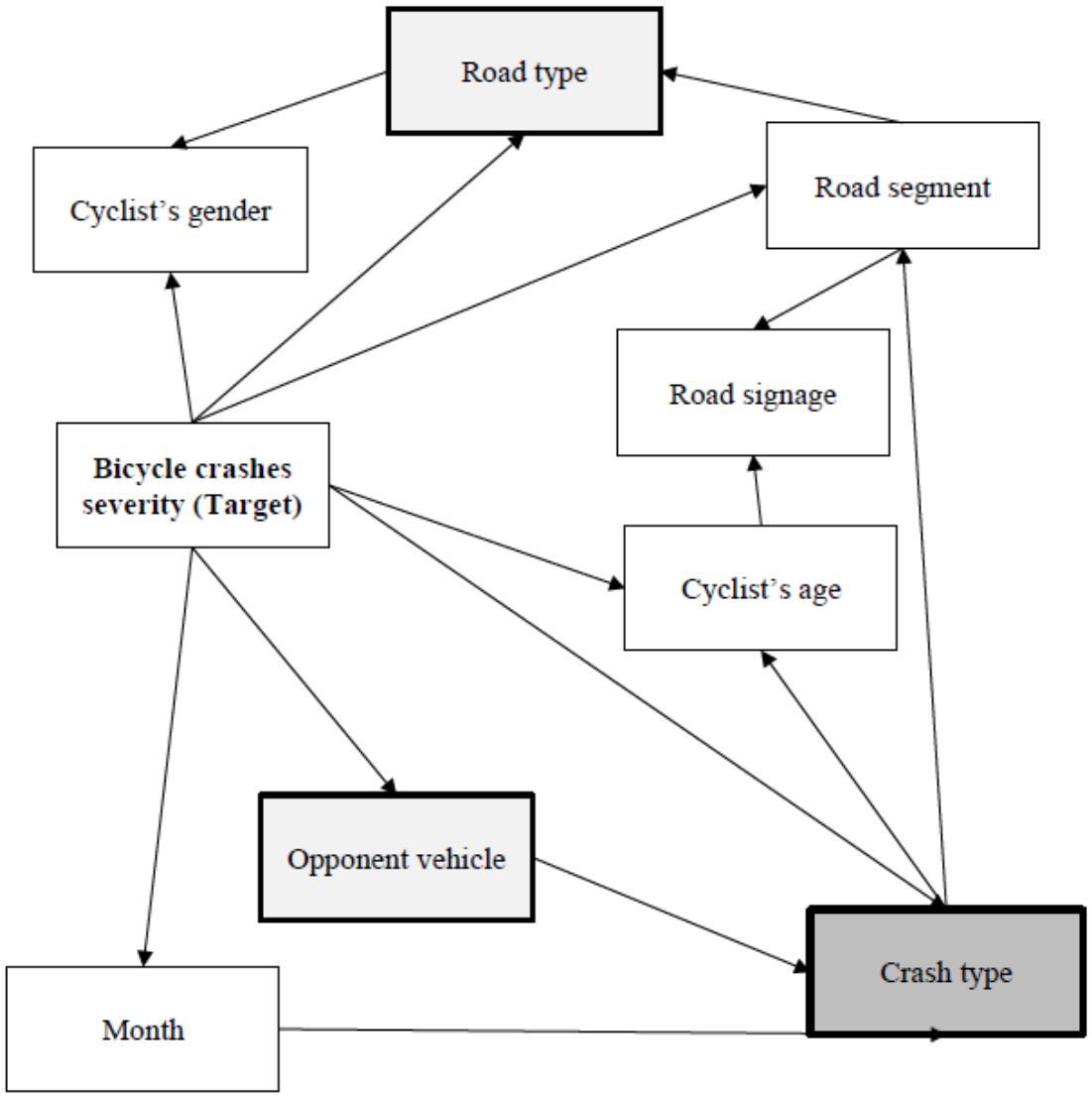
Procedures of BN

Let $U = \{x_1, ..., x_n\}$, $n \geq 1$ be a set of variables and B_p be a set of CPTs, $B_p = \{p(x_i | p_a(x_i), x_i \in U)\}$ where $p_a(x_i)$ is the set of parents of x_i in BN and $i = (1, 2, 3, ..., n)$. A BN represents joint probability distributions P(U):

$$P(U) = \prod_{x_i \in U_p} P(x_i | P_a(x_i))$$

Bayes' theorem can be applied to predict any variable in U given the other variables using $p(x_i|x_j) = \frac{p(x_j|x_i)p(x_i)}{p(x_j)}$. For instance, the classification task consists of classifying a variable y given a set of attribute variables U. A classifier h: $U \rightarrow y$ is a function that maps an instance of U to a value of y. The classifier is learned from a dataset consisting of samples over (U, y).

Example of BN



- Prati et al. (2017) investigated factors related to the severity of bicycle crashes in Italy using the Bayesian network analysis.
- DAG shows the **association** between the severity of bicycle crashes and crash characteristics.
- The network consists of nine nodes, one for the target and one for each predictor.
- The BN model indicates the relative importance of each predictor, using a darker color for more important relationships to the severity of bicycle crashes: crash type (0.31), road type (0.19), and type of opponent vehicle (0.18).

Association Rules vs. Bayesian Networks

- Apriori algorithm is used to calculate the association rules between objects. The typical question asked would be "*what are the frequent items come together*".
- Bayesian networks can infer and learn. Because a Bayesian network is a complete model for its variables and their relationships, it can be used to answer probabilistic queries about them. This process of computing the posterior distribution of variables given evidence is called probabilistic inference. Also, it is necessary to specify for each node X the probability distribution for X conditional upon X's parents through “learning”. The distribution of X conditional upon its parents may have any form.
- Generally speaking, apriori algorithm asks questions on joint probability and all high frequency combinations. It is a “data mining” algorithm. On the other hand, Bayesian network asks questions on conditional probability – given the data, which hypothesis is more likely (probabilistic inference). It is a "machine learning“ algorithm.

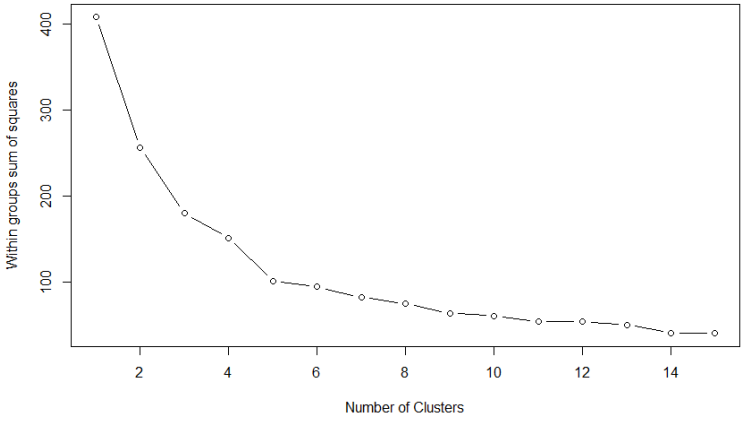
Clustering Analysis (CA)

- An unsupervised learning technique with a principal objective of dividing a dataset into smaller subsets called clusters.
- Based on a heuristic method and tries to maximize the similarity between intra-cluster elements and the dissimilarity between inter-cluster elements (Fraley and Raftery, 2002).
- The two clustering methods commonly used in safety research are K-means Clustering (KC) and Latent Class Clustering (LCC).

K-means Clustering (KC)

- A non-hierarchical, similarity-based clustering method that partitions the observations in *K* clusters, based on the distance of the observations from clusters' means.
- Many algorithms can be used to partition a dataset, including naïve k-means (or Lloyd's algorithm), Forgy and Random Partition, and the Hartigan–Wong method.

  - Prior to running the KC algorithm, a distance function and the value of K need to be specified. A popular choice for the distance function is the *Euclidean distance*.
  - The value of K can be determined visually through a Scree *plot* which exhibits different K values versus the corresponding results in terms of the intra-cluster homogeneity



Latent Class Cluster (LCC)

Latent class cluster (LCC) is a **model-based clustering method** that assumes data from a mixture of probability densities.

$$f(\boldsymbol{y_i}|\theta) = \sum_{k=1}^{K} \pi(C_k) f_k(\boldsymbol{y_i}|C_k, \theta_k)$$

Where
$\boldsymbol{y_i}$ denotes the i^{th} object's scores on a set of observed variables,
K is the number of clusters,
π_k denotes the prior probability of belonging to the latent class k (or membership) and θ are the class-specific parameters.

- The distribution of $\boldsymbol{y_i}$ given the model parameters θ, $f(\boldsymbol{y_i}|\theta)$, is a mixture of class-specific densities $f_k(\boldsymbol{y_i}|C_k, \theta_k)$.
- Maximum-likelihood (ML) and maximum-posterior (MAP) are the two main estimation methods for LCC models, and most software packages use an expectation-maximization (EM) algorithm to find estimates.
- A better way to evaluate and decide the number of clusters in a LCC model is to use information criteria such as AIC, BIC, and CAIC

Example of LCC

Traffic accident segmentation by means of latent class clustering (Depaire et al. (2008))

Table 2
Features and their probability in each cluster

Variable – value	Clu1 (%)	Clu2 (%)	Clu3 (%)	Clu4 (%)	Clu5 (%)	Clu6 (%)	Clu7 (%)
Accident type: collision with a pedestrian	0	99	0	0	0	99	6
Crossroad: crossroad without traffic lights or priority road	74	26	40	21	40	15	5
Crossroad: no crossroad	5	48	1	56	33	76	42
Built-up area: outside built-up area	1	1	1	1	1	0	47
Road type: highway, national, regional or provincial road	25	25	55	23	24	7	99
Age road user 1: 0–18 years old	14	11	16	16	42	96	11
Dynamics road user 2: Road user is not moving	0	11	0	79	74	35	0
Vehicle type road user 1: motorcycle or bicycle	8	0	12	17	90	0	7
Vehicle type road user 1: car	85	0	81	76	9	0	82

4	Traffic accidents between a car and a non-moving second road user	15
5	Traffic accidents with a motorcycle or bicycle	14
6	Traffic accidents with a non-adult pedestrians	10
7	Traffic accidents on highways, national, regional or provincial roads	4

Classification and Regression Trees (CART)

- Introduced by Breiman et al. (1998), CART refers to two types of decision trees: a classification tree and a regression tree.
- A classification tree is an algorithm where the response variable is **categorical**, such as crash injury severity levels, and the algorithm is used to identify the "class" to which a response variable would most likely belong.
- A regression tree refers to an algorithm where the response variable is a **continuous** variable, such as crash frequency or crash rate, and the algorithm is used to predict its value.

CART Procedures

1. Crash data are modeled by a known distribution (e.g., Poisson)

$$L(\boldsymbol{\mu} \mid \mathbf{y}) = \prod_{i=1}^{n} e^{-\mu_i} \mu_i^{y_i} / y_i !$$

where $\mu_i = V^{*}exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik})$.

2. When partitioning a data set recursively, an appropriate splitter needs to be selected through an iterative search for the variable and its specific value from all variables within all the possible levels or values.

$$\Delta D(s,t) = D(t) - D(t_L) - D(t_R)$$

Where t_L and t_R are the left and right child nodes of *t*. D is deviance

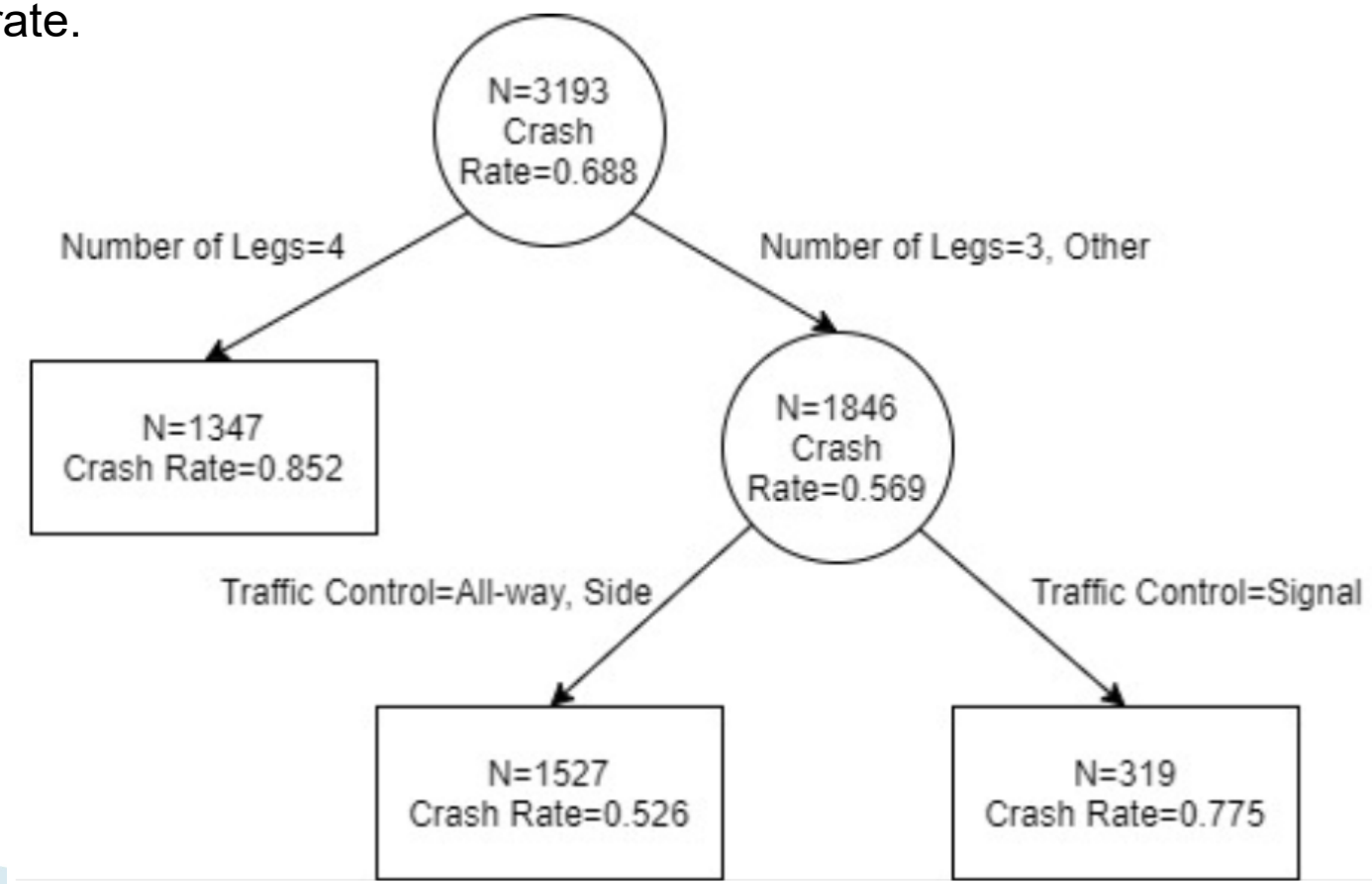
3. The best splitter, *s**, results in the maximum reduction in variability of the dependent variable

$$\Delta D(s^{*},t) = \max_{s \in S} \Delta D(s,t)$$

where *S* is the set of all possible splitters. the reduction at node *t* is the greatest when the deviances at nodes t_L and t_R are smallest.

Example of CART

- In a study by Qin and Han (2008), classification criteria have been developed to categorize similar sites into groups sharing similar attributes to predict crash rate.



CART Implementation

- R 3.5.0 (R Core Team, 2018) includes two available packages, “Tree” and Rpart”, to help build classification and regression trees.
- The key difference between the two packages is the way that missing values are handled during the splitting and scoring processes.
  - In “Tree”, an observation with a missing value for the primary split rule is terminated.
  - “Rpart” offers more flexibility, allowing users to decide how to handle missing values by using surrogates to set up the “usesurrogate” parameter in the rpart.control option.

CART Limitations

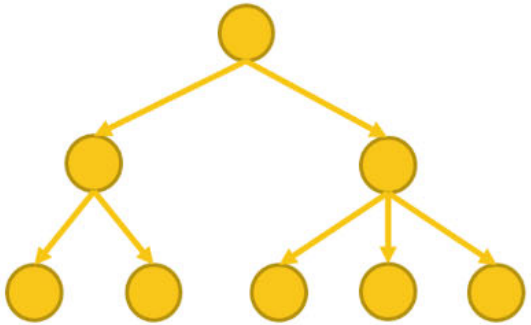
- CART model cannot quantitatively measure the effect of variables on the dependent variable (e.g., injury severity, crash rate) as there is not an estimate coefficient for each variable.
- CART model predicts the outcome based on a single decision tree whose classification accuracy can be unstable due to the data, split variable and complexity of tree change (Chung 2013).
- CART model may cause an overfitting problem (Duncan 1998).

Ensemble Models: Bagging and Boosting

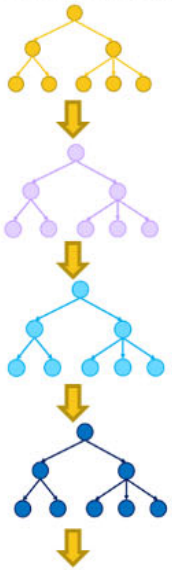
- An alternative to the CART model is ensemble models, which can be used to predict responses by growing **a series of** classification trees.
- The classification trees are grown by randomly selected samples with replacement (i.e., bootstrap samples). Random Forests (RF) and Gradient Boosting Trees (GBT) are representative methods for ensemble learning methods.
- The RF classifier is a type of bootstrap aggregation (i.e., bagged) decision tree in which the ensemble method builds multiple decision trees by repeatedly resampling training data with replacement data. The trees then vote for a consensus prediction.
- GBT builds trees one at a time, and each new tree helps correct mistakes made by the previously trained tree.

Bagging & Boosting

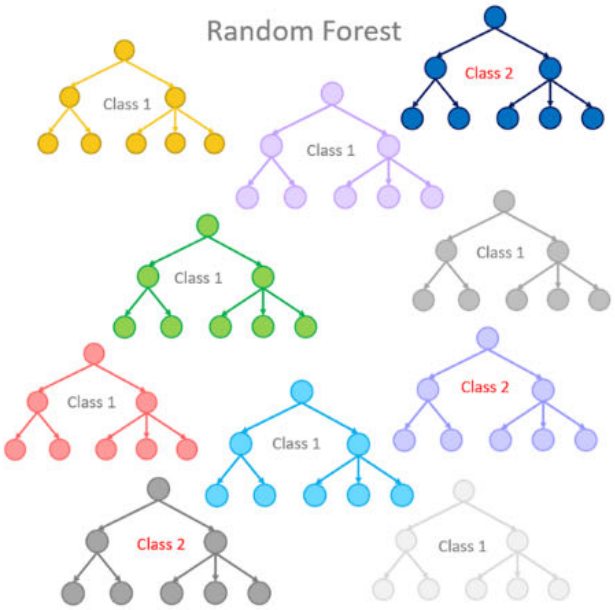
Single Decision Tree



Gradient Boosted Trees



Random Forest



Gradient Boosted Trees (GBT)

- Friedman (2001) introduced the technique as the Gradient Boosting Machine (GBM).
- Use decision trees (especially CART trees) of a fixed size as base learners, so it is often referred to as Gradient Boosted Trees (GBT).
- GBT builds an ensemble of shallow and weak trees sequentially.
- Each new tree in GBT learns and improves on the previously trained tree by applying a higher weight to incorrectly classified observations and a lower weight to correctly classified observations.
- The chance that higher weights will be correctly classified increases when the weak learners are boosted. Hence, the GBT model transforms an ensemble of weak learners into a single strong model and predicts the cases that are difficult to classify.
- The R package implements the “Generalized Boosting Model” method in “gbm”. Interested readers can refer to Elith et al (2017) for details on building GBT with “gbm”.

Algorithm for GBT

Algorithm 12.2: GBT algorithm

1. Initialize $f_0(x)$, which can be set to zero.
2. For n=1,2,3, …, m (number of trees)
   a. For i=1 to k (number of observations), calculate the residual r.
      $r = -\frac{\partial L(y,f(x))}{\partial f(x)}$ where $L(y, f(x)) = (y - f(x))^2$; $f(x) = f_{m-1}(x)$ and $f_{m-1}(x)$ is the basis function for the previous tree (m-1).
   b. Fit a decision tree to r to estimate γ_n
   c. Estimate β_n by minimizing $L(y_i, f_{n-1}(x) + \beta_n b(x; \gamma_n)$
   d. Update $f_n(x) = f_{n-1}(x) + \beta_n b(x; \gamma_n)$
3. Calculate $f(x) = \sum_{n=1}^{m} \beta_n b(x; \gamma_n)$

Where , a basis function $f(x)$ describes a response variable y in a function of the summation of weighted basis functions for individual trees; $b(x; \gamma_n)$ is the basis function for individual tree n; m is the total number of trees; γ_n is the split variable; and, β_n is the estimated parameter that minimizes the loss function, $L(y, f(x))$.

Pros and Cons of GBT

- Handle different types of predictor variables and accommodate missing data.
- Fit complex nonlinear relationships and automatically account for interactions between predictors.
- Since boosted trees are built by optimizing an objective function, GBT can be used to solve almost all objective functions as long as the gradient functions are available.
- Boosting focuses step by step on difficult cases is an effective strategy for handling unbalanced datasets because it strengthens the impact of positive cases.
- However, GBT training generally takes time because trees are built sequentially, and each tree is built to be shallow.
- Therefore, the quantitative effect of each variable on the response variable, such as crash injury severity, may not be available.

Random Forests (RF)

- Breiman (2001) proposed this method as a prediction tool that uses a collection of tree-structured classifiers with independent, identically distributed random vectors.
- R package “randomForest” which is based on Breiman’s (2001) method
- “Randomness” helps to make the model more robust than a single tree and less likely to overfit the training data.
- Unlike GBT whose trees are built sequentially, RF builds an ensemble of deep independent trees.
- However, RF do not identify whether a variable has a positive or negative effect on the response variable. Hence, RF are often used to rank the importance of variables as a screening method for selecting input variables for other models such as a logistic regression.
- For a categorical variable with multiple levels, RF are biased in favor of the attribute values with more observations and may produce unreliable variable importance scores.
- Also, a large quantity of trees may make the algorithm slow for real-time prediction.

Algorithm and Implementation

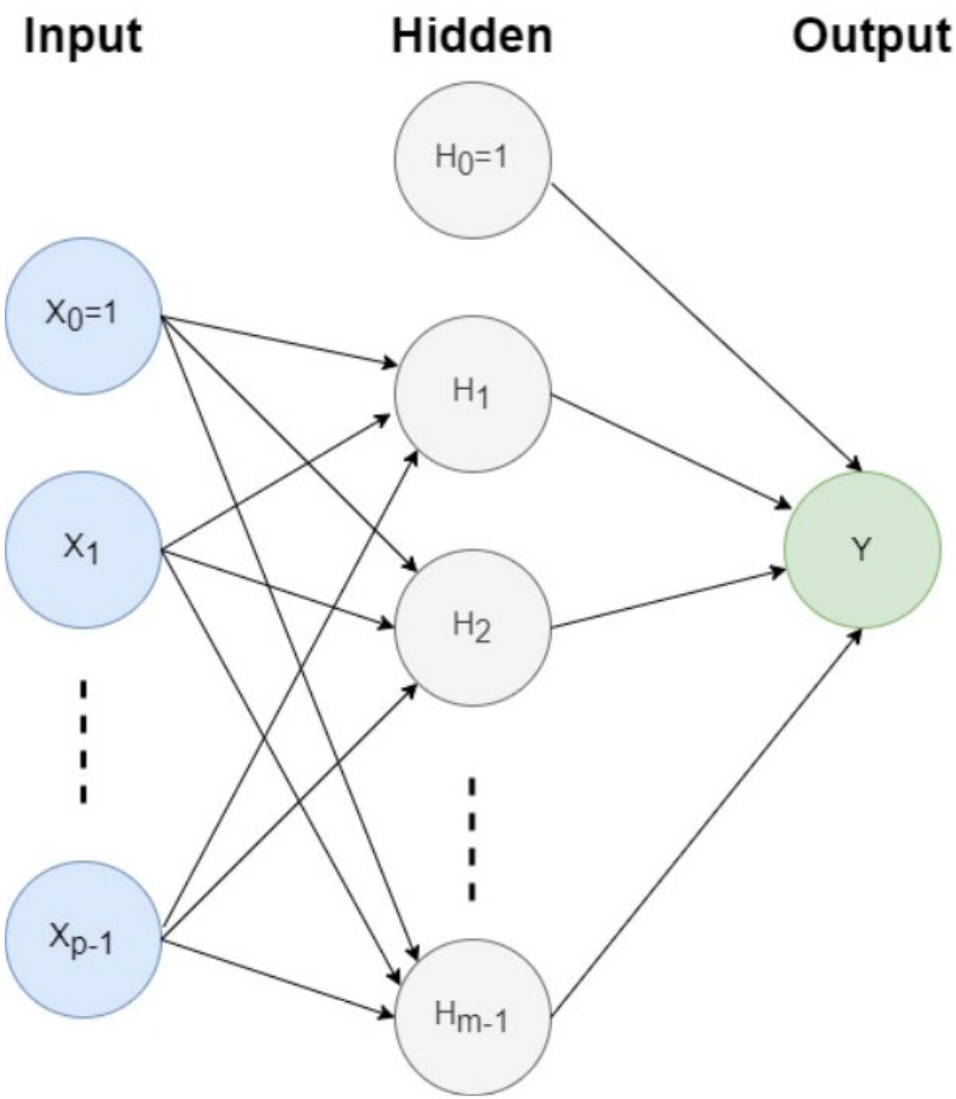
Algorithm 12.1: Random Forests algorithm

1. Select a bootstrap sample.
2. Grow a classification tree to fit the bootstrap sample so that the variable can be selected only from a small subset of randomly selected variables for each split in the classification tree.
3. Predict the response variable for the samples not selected in the bootstrap sample (i.e., out-of-bag samples) by using the classification tree in Step 2. The predicted category of the response variable is the category with the highest proportion of samples.
4. Compare the observed and predicted categories of the response variable to calculate the rate of incorrect classification of the sample (the number of misclassified samples over total number of samples) for each tree. This rate is defined as the misclassification rate (r_b).
5. For each predictor variable i, permute the value of the variable in the out-of-bag samples. Predict the response variable by using the classification tree in Step 2 to calculate the new misclassification rate of the tree (r_{ai}). The importance score for variable i is computed on the basis of the difference between the misclassification rates before and after the permutation $[(r_{ai} - r_b)/r_b]$. A higher difference between the two misclassification rates increases the importance score, meaning the variable importance is higher. The importance score for each variable is updated as more trees are trained to the out-of-bag samples.
6. Repeat Steps 1 to 5 until enough trees are grown by using different bootstrap samples. Calculate the average importance score for each variable in different trees.

Neural Network

- The artificial neural network (ANN) is a machine learning technique used to model the response in a large dataset as a nonlinear (activation) function of linearly combined predictors.
- ANN is a means to effectively discover new patterns and correctly classify data or make forecasts.
- The output signal from one neuron can be used as an input for other neurons, thus effectively modeling and solving complex problems through a network of multiple neurons and multiple layers.
- Several types of ANNs are described in the following slides.

Multilayer Perceptron (MLP) Neural Network



- The Feed Forward Neural Network (FNN) effectively solves multivariate nonlinear regression and classification problems.
- MLP neural network is a class of FNN in which neurons of the same layer are not connected to each other, but to the neurons of the preceding and subsequent layers.
- An output of one hidden layer serves as an input to the subsequent layer in the form of the activation function of a weighted summation of the outputs of the last hidden layer.
- The weights can be determined by solving the optimization problem, or minimizing a given cost function.
- The algorithm most commonly used to determine the weights is back-propagation.

Procedures of MLP

In Fig. 12.3, the MLP model, Xs are the independent variables (Indicators), Hs are the hidden nodes, and Y is the dependent variable. αs are the estimated coefficients between hidden nodes and indicators, and *β*s are the estimated coefficients between hidden nodes and dependent variables. The model can be described as:

$$Y = g_Y(\beta_0 + \beta_1 \mathrm{H}_1 + \cdots + \beta_{m-1}\mathrm{H}_{m-1}) + \boldsymbol{\varepsilon}$$

where $\mathrm{H}_i = \mathrm{g}_H\left(\sum_j \alpha_{i,j}\mathrm{X}_j\right)\ j = 0, 1, \ldots p-1$; ε is the error term; and g_Y and g_H are the activation functions. In ANN, the activation function of a node defines the output of that node given an input. Fig. 12.4. shows common activation functions such as the radial basis function (e.g., Gaussian), the sigmoidal function (e.g., logistic), the hyperbolic function (e.g., tanh), and the rectified linear unit (ReLU). For example, if the activation function is the logistic function: $g(z) = (1 + e^{-z})^{-1}$, the model can be transformed to:

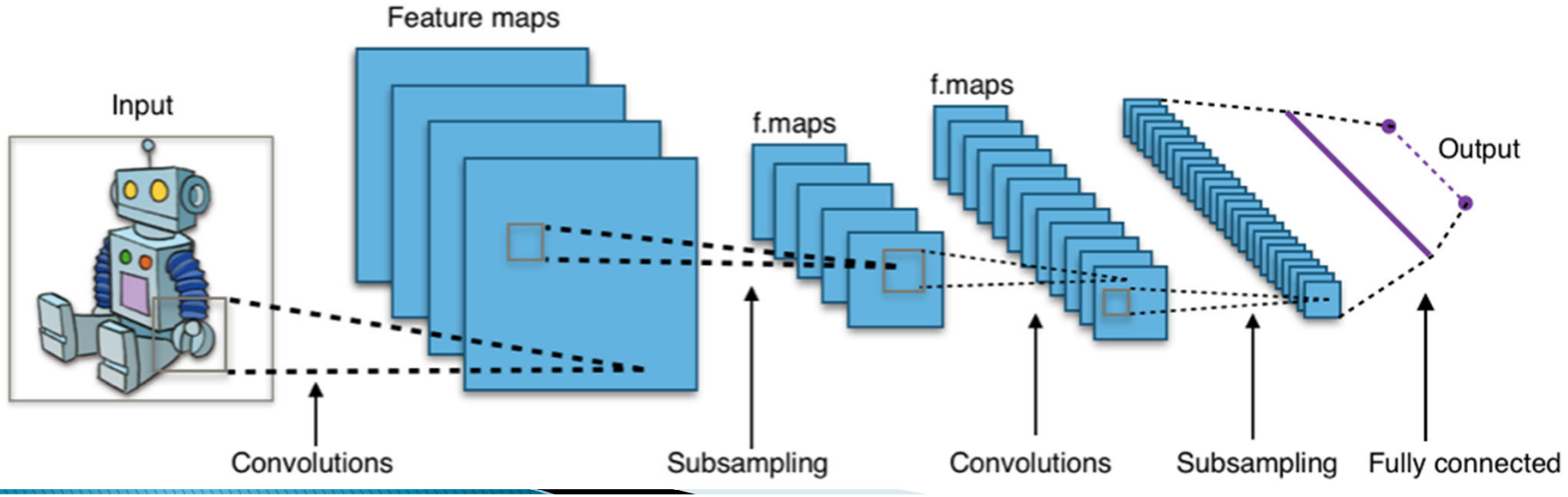
$$Y = [1 + \exp\,[-\beta_0 - \textstyle\sum_{n=1}^{m-1} \beta_n[1 + \exp(-\alpha_{n,j}\mathrm{X}_j)]^{-1}]]^{-1} + \boldsymbol{\varepsilon} = f(\boldsymbol{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}) + \boldsymbol{\varepsilon}$$

Multi-class Classification Problem

- ANN models, like other classification models, have the multi-class classification problem, meaning they ignore less-represented categories in order to improve the model's overall accuracy.
- The multi-class classification problem is a problem for injury severity studies since the data are highly skewed due to the presence of fewer high-severe injury crashes and more less-severe injury crashes.
- One solution is reducing the multi-class problem into multiple two-class (binary) classification problems.
- Another technique is resampling: oversampling less-representative classes, or undersampling overly-representative classes, or using ensemble methods (e.g., Bootstrap aggregating).
- Jeong et al. (2018), used undersampling, oversampling, and ensemble methods (majority voting and bagging) to classify motor vehicle injury severity. Yuan et al. (2019) used the Synthetic Minority Over-sampling Technique (SMOTE) algorithm for unbalanced classification problems to predict real-time crash risk in their long short-term memory RNN.

Convolutional Neural Networks (CNN)

- Convolutional neural networks (CNN, or ConvNet) are Deep Learning neural networks that are commonly applied to image analyses.
- The name “convolutional” indicates that the algorithm employs a mathematical operation called “convolution”.
- CNN typically consists of convolutional layers, pooling layers, and fully connected layers. The convolutional layer is the core building block of CNN.

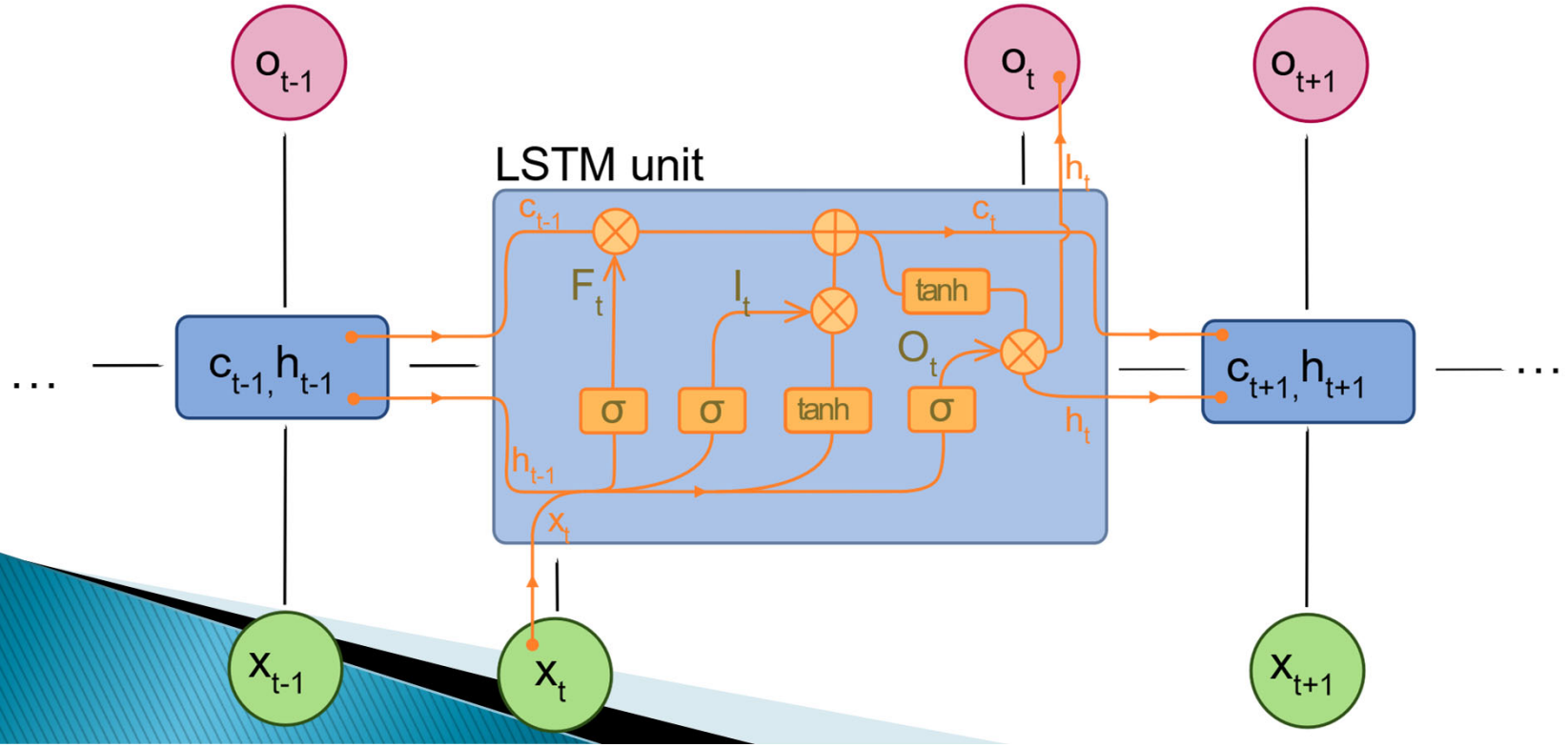


Procedures of CNN

- As shown in the figure, a convolutional layer creates a filter that slides over the image spatially, resulting in a feature map.
- Various filters can produce many separate feature maps which are stacked to generate volume. The results are then passed to the next layer.
- Pooling the layers reduces the size of convoluted feature by combining the outputs of node clusters at one layer into a single node in the next layer. Pooling is a form of non-linear down-sampling where max pooling is the most common among several non-linear functions.
- Finally, after several convolutional and max pooling layers, the important features of an image can be understood.
- The matrix that represents the extracted features will be flattened and fed into a traditional MLP neutral network for classification purposes.

Long Short-Term Memory - Recurrent Neural Networks (LSTM-RNN)

- RNN is a very important variant of neural networks that is heavily used in Natural Language Processing.
- In RNN, connections between nodes form a directed graph following a temporal sequence, allowing temporal patterns in the data to be modeled.
- RNNs can retain a “memory” that is captured in the time-series data.



Components of RNN

- The **input gate** decides which value from the input should be used to modify the memory. For example, a sigmoid function decides which values to let through, and a *tanh* function assigns weights to the passing values for their levels of importance, from -1 to 1.
- The **forget gate** determines which memories the cell can forget. For example, a sigmoid function outputs a number between 0 and 1 for each number in the cell state C_{t-1} based on the previous state (h_{t-1}) and the content input (X_t).
- The **output gate** yields the output based on the input and the memory of the cell. For example, functioning similar to the input gate, a sigmoid function decides which values to let through. A *tanh* function gives weights to the passing values for their levels of importance ranging from -1 to 1 and is then multiplied with the output of the sigmoid function.

Support Vector Machines (SVM)

- A support vector machine (SVM) is a machine learning approach originally developed by Vapnik et al. (Cortes. and Vapnik; 1995; Vapnik, V., 1998).
- SVM includes a set of supervised learning methods that can be used for classification and regression analysis.
- A simple two-class classification problem is illustrated in Fig. 12.7.

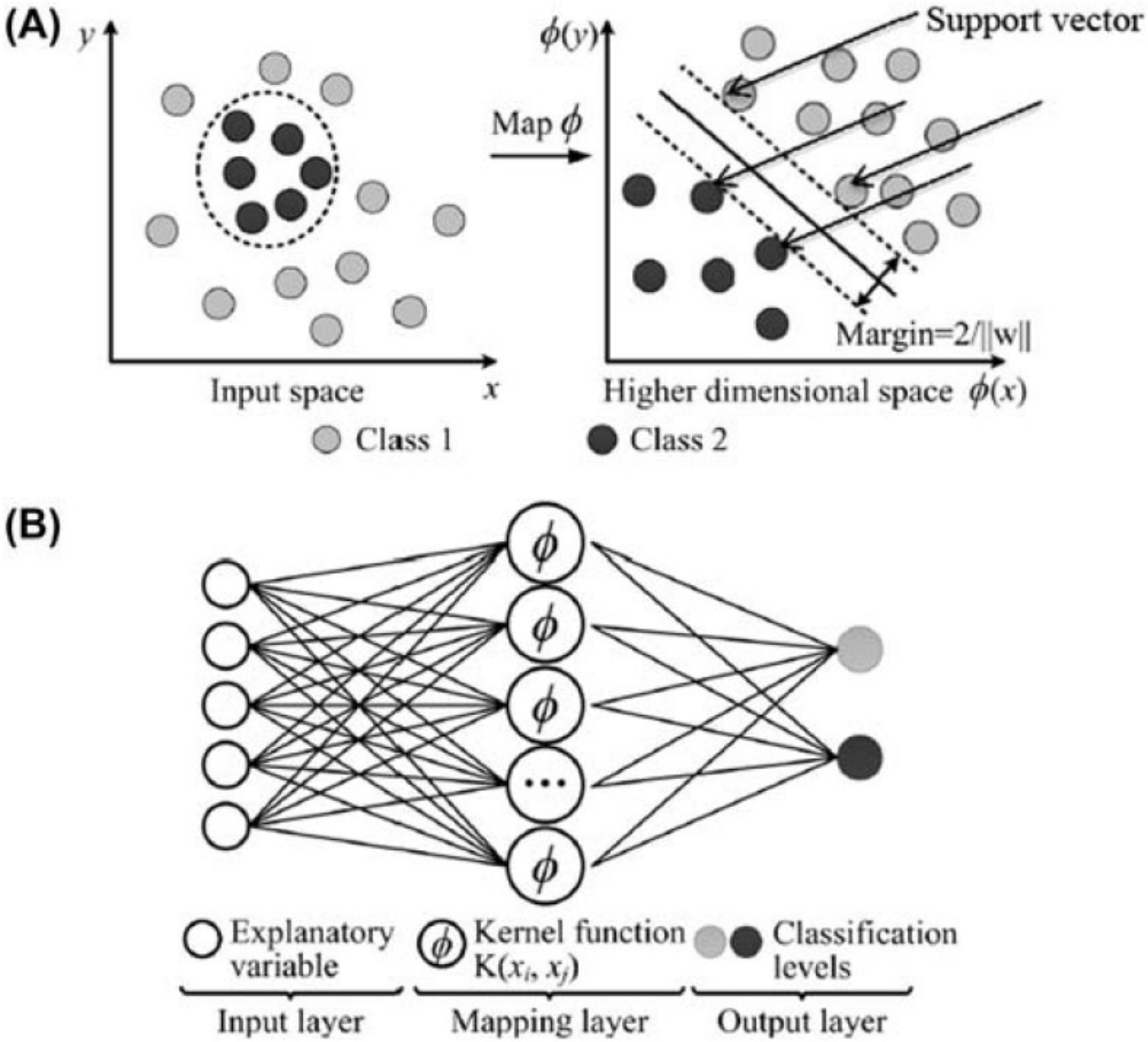


FIGURE 12.7 Classification of SVM models (Li et al., 2012).

A Simple Two-class Classification Problem

- First, the input data points are mapped from data space to a high dimensional feature space using a non-linear kernel function such a Gaussian kernel.
- The SVM model then constructs two separating hyperplanes (see dashed line in Fig. 12.7(a)) in the high dimensional space to separate the outcome into two classes so that the distance between them is as large as possible.
- The region bounded by the two separating hyperplanes is called the “margin”, and the optimal separating hyperplane is in the middle (see the solid line in Fig. 12.7(a)).
- The idea is to search for the optimal separating hyperplane by maximizing the margin between the classes’ closest points.
- The points lying on the boundaries are called support vectors. Fig. 12.7(b) represents a typical neural network with one input layer, one hidden layer and one output layer.

Procedures of SVM

For the two-class classification problem, given a training set of instance-label pairs (x_i, x_j), the SVM model aims to solve the optimization problem in Equation 12.17 (Cortes and Vapnik, 1995).

$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$

Subject to: $y_i\left(w^T \phi(x_i) + b\right) \geq 1 - \xi_i, \xi_i \geq 0$

where ξ_i is a slack variable that measures misclassification error; *C* is a regularization parameter which is the penalty factor to errors (e.g., a large parameter C value indicates a small margin, and vice versa). The coefficient C is still undetermined, but this optimization problem can be solved using the Lagrange multiplier:

$$\max min \left\{ \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \left[y_i \left(w^T \phi(x_i) + b \right) - 1 + \xi_i \right] - \sum_{i=1}^{N} \beta_i \xi_i \right\}$$

where $\alpha_i, \beta_i > 0$ are Lagrange multipliers. The max sign means that among all hyperplanes separating the data, one is unique in yielding the maximum margin of separation between the classes. $\Phi(x_i^T)\Phi(x_j)$ is the kernel function.

SVM Applications in Highway Safety Analysis

- SVM's superior performance has led to its popularity in highway safety analysis, for injury severity classification and for crash count prediction.
- Li et al. (2012) applied the C-SVM models to predict injury severity of crashes at freeway diverge areas.
  - The SVM model was better at predicting crash injury severity when compared with the ordered probit (OP) model
  - However, the performance of the SVM model depends highly on the learning procedure, which contains functional mapping and parameter selection.
  - The authors suggested using kernel functions other than the basic RBF kernel to improve the model performance.
- Li et al. (2008) also applied *v*-SVM to predict motor vehicle crashes on frontage roads in Texas.
  - The SVM model results were compared with traditional NB regression models, and several sample sizes were evaluated for examination of data fitting and model prediction capabilities.
  - They found that the SVM models were consistently lower with regard to the values of Mean Absolute Deviation (MAD) and Mean Absolute Percentage Error (MSPE), for all sample sizes than those for NB regression models.

Implementation in R

- Statistical software R has several packages for dealing with ANNs, such as “neuralnet”, “nnet” and “RSNNS”.
- However, not all packages include the ability to plot the function, which allows readers to visualize the neural networks.
- Additionally, the ability to take separate or combined x and y inputs as data frames or as a formula may also not be included.
- Image recognition in statistical software R 3.5.0 (R Core Team, 2018) uses deep CNN in the “MXNet” package.
- The R 3.5.0 (R Core Team, 2018) package “rnn” implements LSTM, Gated Recurrent Unit (GRU) and Vanilla RNN models.
- The “keras” R package, an open-source neural-network library written in Python, may be another option. “keras” was developed to enable fast experimentation and supports both convolution based networks and recurrent networks.

Implementation in R (cont'd)

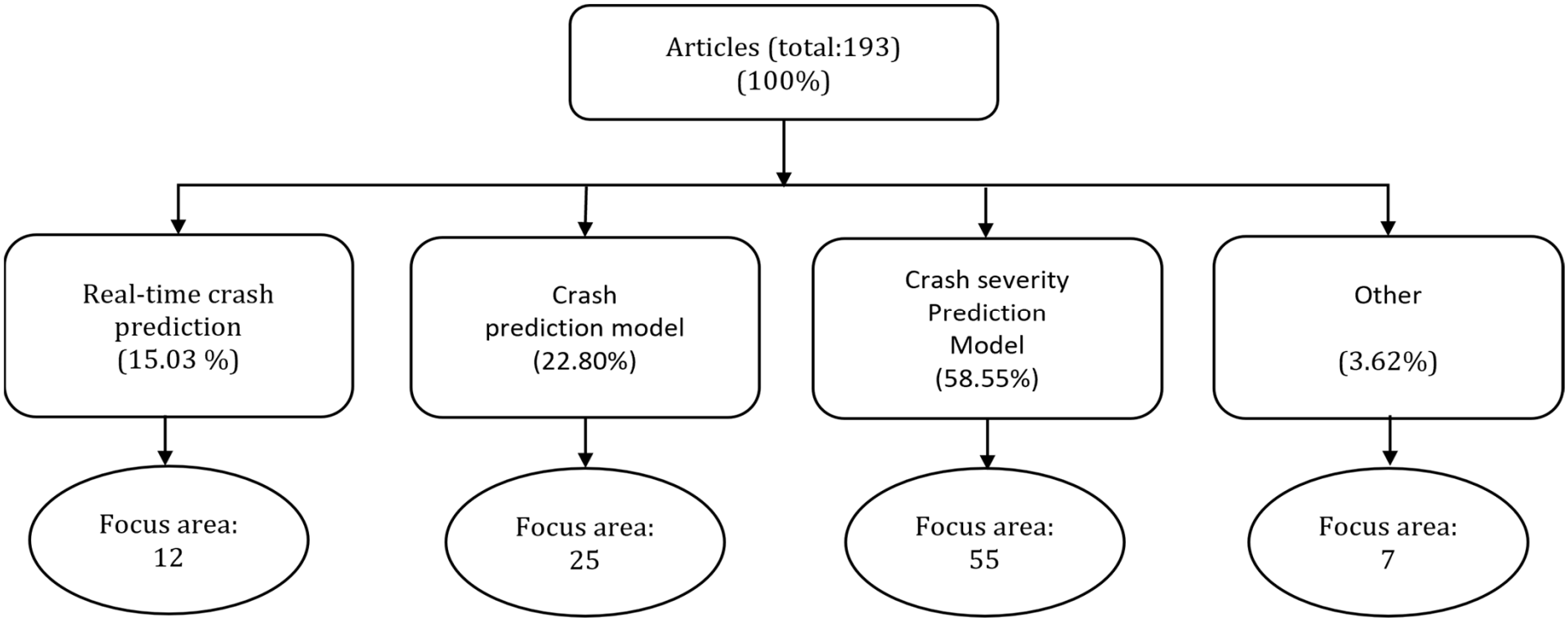
- The R interface for 'H2O', is another choice. H2O is a fully open source machine learning platform that offers parallelized implementations of many supervised and unsupervised machine learning algorithms such as Generalized Linear Models, Gradient Boosting Machines (including XGBoost), Random Forests, Deep Neural Networks (Deep Learning), Stacked Ensembles, Naive Bayes, Cox Proportional Hazards, K-Means, PCA, Word2Vec, as well as a fully automatic machine learning algorithm (AutoML) (https://cran.r-project.org/web/packages/h2o/index.html).
- The R interface to "libsvm" is in package "e1071" where svm() includes C-classification, *v*-classification, one-class-classification (novelty detection), ε-regression and v-regression, svm() also includes: linear, polynomial, radial basis function, and sigmoidal kernels, formula interface, and k-fold cross validation. For further implementation details on libsvm, see Chang & Lin (2001).
- Software package development is an evolving process, so readers are encouraged to check regularly for new developments and compare the differences between these options in R.

Sensitivity Analysis

- One of the methods safety practitioners have used to address this concern is the sensitivity analysis (Delen et al. 2006, Li et al, AAP 2008, Li et al., 2012).
- The sensitivity analysis studies how the uncertainty in the output of a mathematical model can be attributed to different sources of uncertainty in its input.
- The sensitivity analysis can be used to measure the relationship between the input variables and output of a trained neural network model (Principe et al., 2000).
- In the process of performing a sensitivity analysis, the neural network learning ability is disabled so that the network weights are not affected. Each input variable of a black-box model (e.g., ANN, SVM) is perturbed by a user-defined amount, with the other variables being fixed at their respective means or medians.
- The results before and after the perturbation of each input variable are recorded, and the impacts of each input variable on the output are calculated.
- For example, in the crash injury severity level prediction, the percent change of each severity level by one unit change of each input variable can be estimated.

A Review of ML in Highway Safety Research in the Past Decade (2012-2021)

Research Topics and Focus Areas

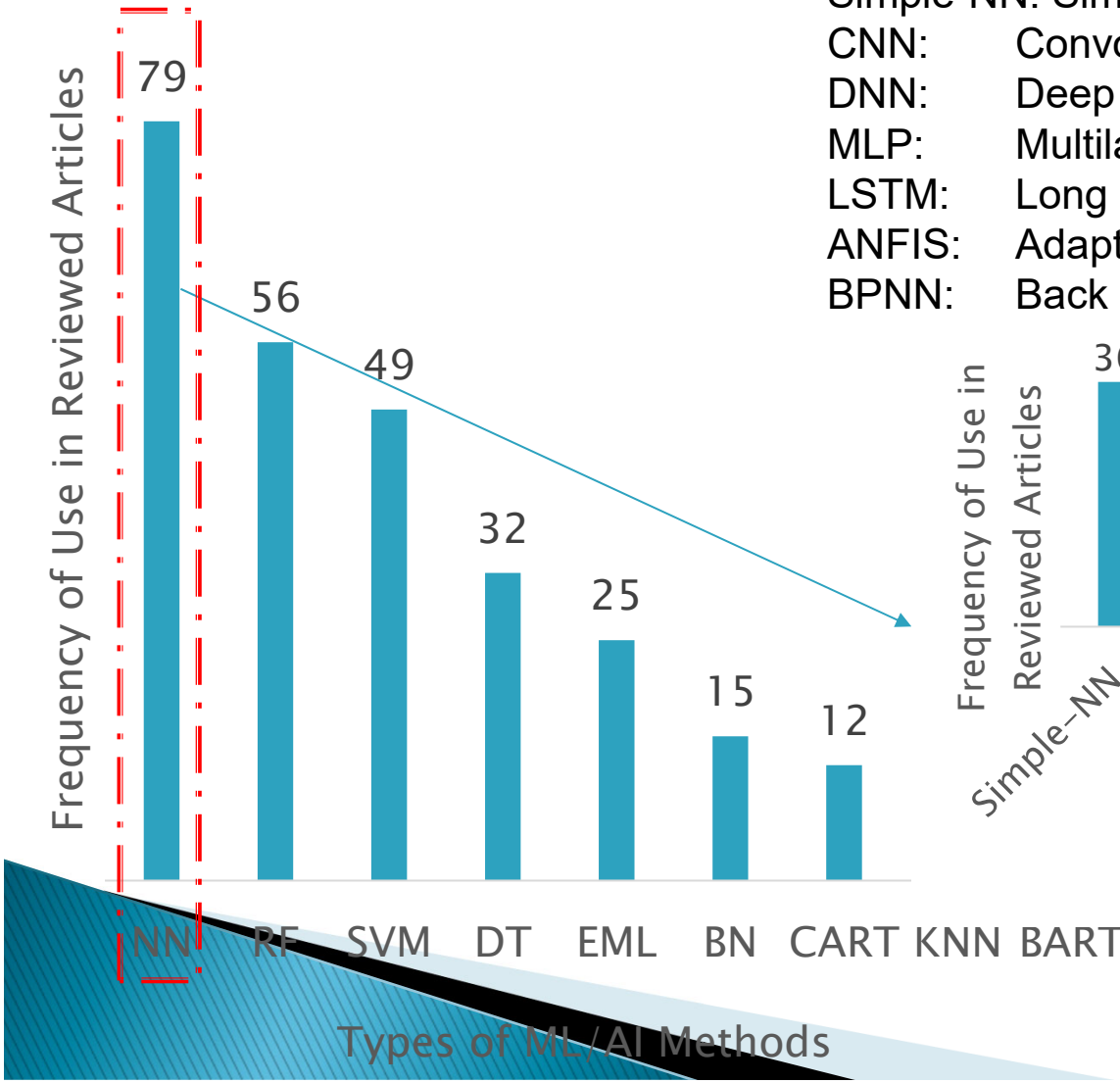


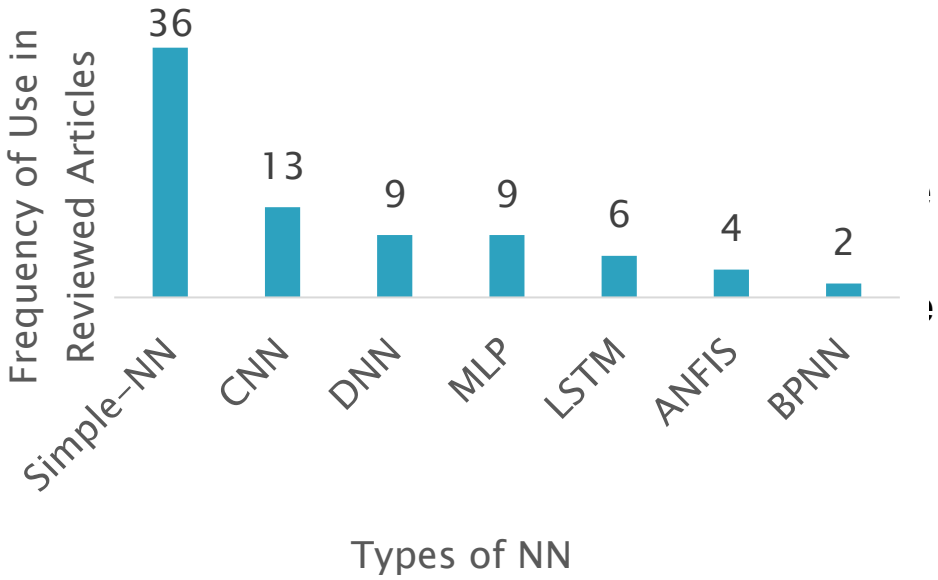
Other: Driver Behavior and Data Augmentation

- There are three subcategories, which are distracted driver crash analysis, driver-vehicle volatilities, and rural road crash analysis.

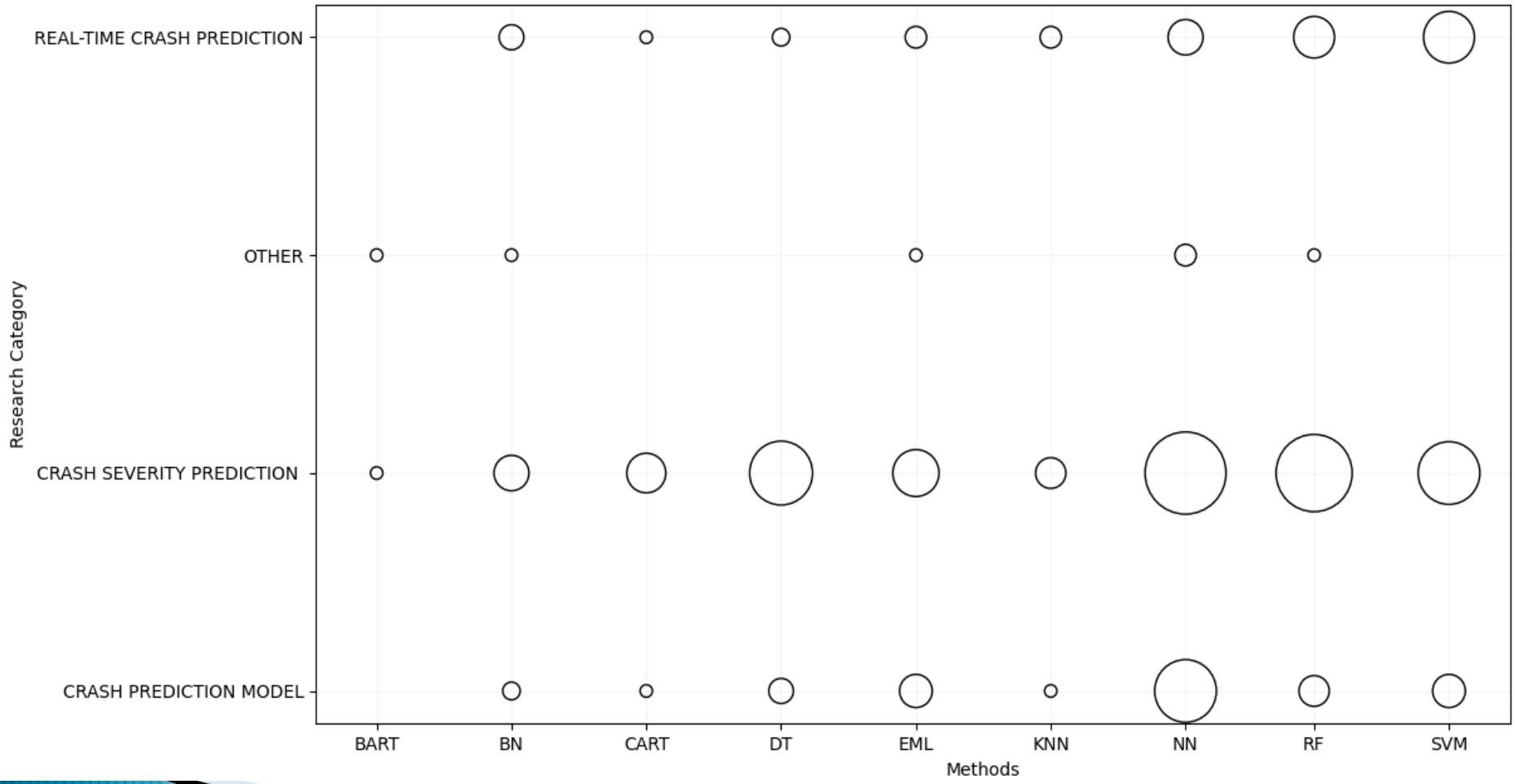
Method Popularity

Simple-NN: Simple-Neural Network
CNN: Convolutional Neural Network
DNN: Deep Neural Network
MLP: Multilayer Perceptron
LSTM: Long Short-Term Memory
ANFIS: Adaptive Neuro Fuzzy Inference System
BPNN: Back Propagation Neural Network

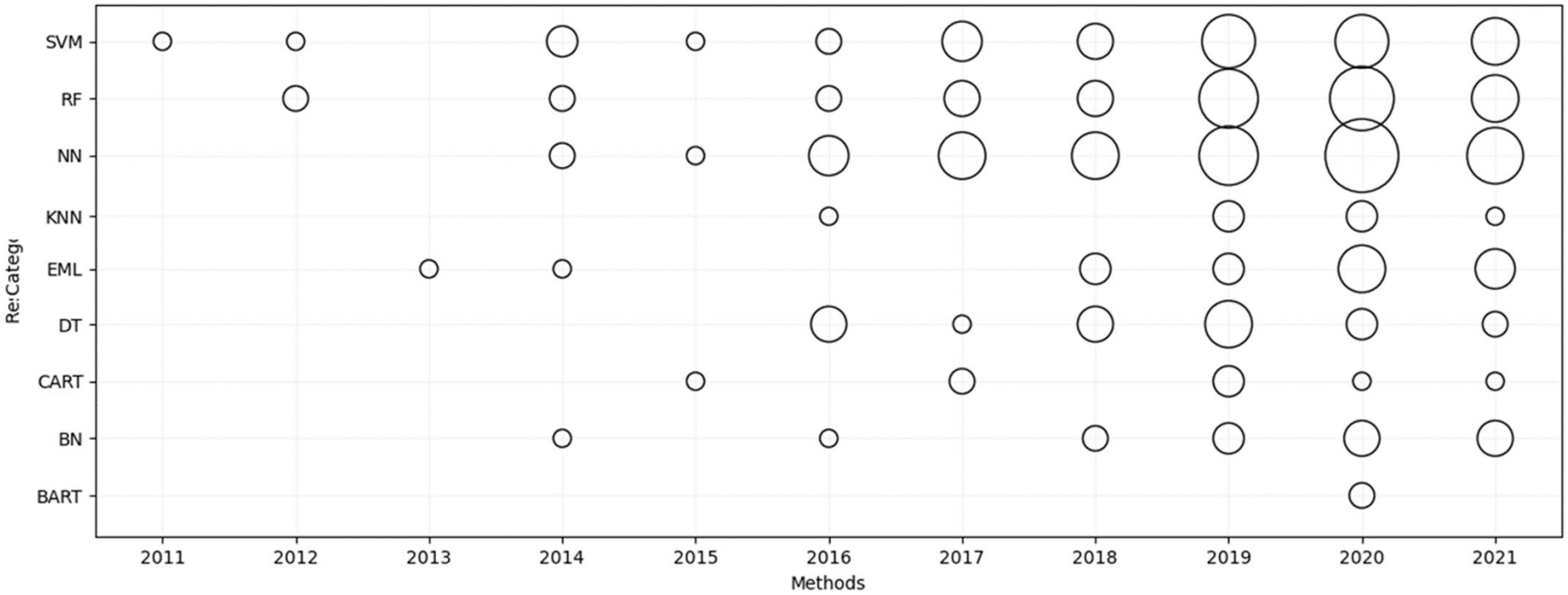




Research Topics, Focus Areas, and Methods

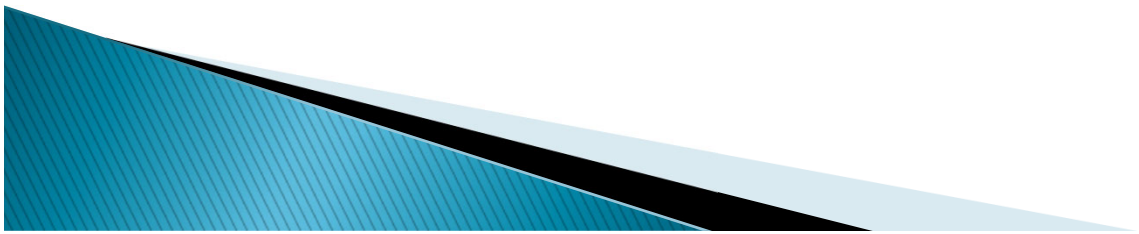


Temporal Trends among/of the Research Topics and Methods



Proposed vs. Outcome of Existing Work

- (Proposed) Link a large amount of data or to handle relevant big data and many input variables. (Outcome), in most cases the datasets utilized for demonstration this purpose are not considered big data.
- (Proposed) Improved model prediction performance. (Outcome), the magnitude of such improvements is marginal.
- (Proposed) Capture complex, nonlinear, or heterogeneous relationships in the safety data. (Outcome), the issue of interpretability from ML/AI remains critical.



Future Research Directions and Recommendations

- Research using real-time crash prediction models to explore the determinants of crashes dominated this field, while very limited studies focused on investigating crash root causes and developing safety interventions.
- Driver behavior data (e.g., naturalistic driving study (NDS)) in couple with ML/AI methods continues to accelerate behavioral safety studies.
- Few studies discussed the validation and transferability issue of their ML/AI models.
- Lack of guidelines and standards in the collection, management, and integration of those emerging traffic crash relevant data.
- Call for more and greater collaborations between academia and industry and inter-disciplines. Our study shows only 7% of the studies have an industry partner; and only 22% have a partner from other disciplinaries than transportation engineering.

References

1. Abdelwahab, H.T., Abdel-Aty, M.A., 2002. Artificial neural networks and logit models for traffic safety analysis of toll plazas. Transportation Research Record: Journal of the Transportation Research Board 1784, 115–125.
2. Breiman, L. Random Forests. Machine Learning, Vol. 45, No. 1, 2001, pp. 5–32.
3. Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Chapman & Hall/CRC, Boca Raton, Fla., 1998.
4. Chang, C.-C., Lin, C.-J., 2001. Training v-Support Vector Classifiers: theory and algorithms. Neural Computation 13 (9), 2119–2147.
5. Chang, C.-C., Lin, C.-J., 2007. LIBSVM: A Library for Support Vector Machines, www.csie.ntu.edu.tw/ cjlin/libsvm.
6. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.
7. Cortes, C., Vapnik, V., 1995. Support-vector network. Machine Learning 20, 273–297.
8. Cristianini, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.
9. Delen, D., Sharda, R., & Bessonov, M. (2006). Identifying significant predictors of injury severity in traffic accidents using a series of artificial neural networks. Accident Analysis & Prevention, 38(3), 434-444.
10. Depaire, B., Wets, G., & Vanhoof, K. (2008). Traffic accident segmentation by means of latent class clustering. Accident Analysis & Prevention, 40(4), 1257-1266.
11. Elith, J. and Leathwick, J., Boosted Regression Trees for ecological modeling, https://cran.r-project.org/web/packages/dismo/vignettes/brt.pdf, 2017
12. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 1189–1232
13. Fraley, C., Raftery, A.E., 2002. Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association 97 (458), 611–631.
14. Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM.
15. Hagenaars, J. A., & McCutcheon, A. L. (Eds.). (2002). Applied latent class analysis. Cambridge University Press.
16. Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100-108.
17. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
18. Iranitalab, A., & Khattak, A. (2017). Comparison of four statistical and machine learning methods for crash severity prediction. Accident Analysis & Prevention, 108, 27-36.
19. Jeong, H., Y. Jang, P. J. Bowman, and N. Masoud, "Classification of motor vehicle crash injury severity: A hybrid approach for imbalanced data," Accident Analysis & Prevention, vol. 120, pp. 250–261, 2018
20. Kecman, V., 2005. Support vector machines—an introduction. In: Wang, L. (Ed.), Support Vector Machines: Theory and Applications. Springer-Verlag, Berlin, Heidelberg, New York, pp. 1–48.
21. Kononov, J., Lyon, C., and Allery, B., (2011). Relation of Flow, Speed, and Density of Urban Freeways to Functional Form of a Safety Performance Function. Transportation Research Record: Journal of the Transportation Research Board (2236), 11–19.
22. Krizhevsky et al., 2012, Krizhevsky, A Sutskever, I, and Hinton, G.E., ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems (NIPS), pp. 1097-1105
23. Lee, C., & Li, X. (2015). Predicting driver injury severity in single-vehicle and two-vehicle crashes with boosted regression trees. Transportation research record, 2514(1), 138-148.
24. Li, X., Lord, D., Zhang, Y., & Xie, Y. (2008). Predicting motor vehicle crashes using support vector machine models. Accident Analysis & Prevention, 40(4), 1611-1618.
25. Li, Z., Liu, P., Wang, W., & Xu, C. (2012). Using support vector machine models for crash injury severity analysis. Accident Analysis & Prevention, 45, 478-486.
26. Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.
27. Liang, F. (2003). An effective Bayesian neural network classifier with a comparison study to support vector machine. Neural Computation, Vol. 15(8), pp. 1959-1989.
28. Liang, F. (2005). Bayesian neural networks for nonlinear time series forecasting. Statistics and Computing, Vol. 15(1), pp. 13-29.
29. Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors, 17(4), 818.
30. McLachlan, G.J., Peel, D., (2000). Finite Mixture Models. Wiley, New York.
31. Neapolitan, R. E. (2004). Learning Bayesian Networks (Vol. 38). Upper Saddle River, NJ: Pearson Prentice Hall. Prevention, Vol. 61, 2013, pp. 107–118.
32. Prati, G., Pietrantoni, L., & Fraboni, F. (2017). Using data mining techniques to predict the severity of bicycle crashes. Accident Analysis & Prevention, 101, 44-54.
33. Principe, J.C., Euliano, N.R., Lefebre, W.C., 2000. Neural and Adaptive Systems: Fundamentals Through Simulations. John Wiley and Sons, New York.
34. Qin, X., & Han, J. (2008). Variable selection issues in tree-based regression models. Transportation Research Record, 2061(1), 30-38.
35. Sch¨olkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L., 2000. New support vector algorithms. Neural Computation 12, 1207–1245.
36. Suykens, J.A.K., Gestel, T.V., Brabanter, J.D., Moor, B.D., Vanderwalle, J., 2002. Least Squares Support Vector Machines. World Scientific Publishing Co. Pte. Ltd., Singapore.
37. Vapnik, V. (1998). Statistical learning theory. New York: Wiley.
38. Vapnik, V., Golowich, S. E., & Smola, A. J. (1997). Support vector method for function approximation, regression estimation and signal processing. In Advances in neural information processing systems (pp. 281-287).
39. Wang, K., [illegible] Qin, X. (2015). Exploring driver error at intersections: key contributors and solutions. Transportation research record, 2514(1), 1-9.
40. Xie, Y., D. Lord, and [illegible] (2007) Predicting Motor Vehicle Collisions using Bayesian Neural Networks: An Empirical Analysis. Accident Analysis & Prevention, Vol. 39, No. 5, pp. 922-933.