A Continuous Learning Framework for Freeway Incident Detection

Srinivas Peeta
Debjit Das

School of Civil Engineering
Purdue University
West Lafayette, IN 47907

(765) 494-2209
(765) 496-1105 (FAX)
peeta@ecn.purdue.edu
das@ecn.purdue.edu

ABSTRACT

Existing freeway incident detection algorithms predominantly require extensive off-line training and calibration precluding transferability to new sites. Also, they are insensitive to demand and supply changes in the current site without recalibration. We propose two neural network based approaches that incorporate an on-line learning capability, thereby ensuring transferability, and adaptability to changes at the current site. The Least Squares technique and the Error Back Propagation algorithm are used to develop on-line neural network trained versions of the popular California algorithm and the more recent McMaster algorithm. Simulated data from the INTRAS model is used to analyze the performance of the neural network based versions of the California and McMaster algorithms over a broad spectrum of operational scenarios. The results illustrate the superior performance of the neural net implementations in terms of detection rate (DR), false alarm rate (FAR), and time to detection (TTD). Of implications to current practice, they suggest that just introducing a continuous learning capability to commonly employed detection algorithms in practice such as the California algorithm enhances their performance with time in service, allows transferability, and ensures adaptability to changes at the current site. An added advantage of this strategy is that existing traffic measures used (such as volume, occupancy, etc.) in those algorithms are sufficient, circumventing the need for new traffic measures, new threshold parameters, and/or variables requiring subjective decisions.

KEYWORDS

Incident detection, Neural Networks, Continuous learning.

TRB Paper Number: 980962
Word Count: 4738
Number of Figures: 11

INTRODUCTION
Automatic incident detection is an important component of a freeway traffic management system. Though several approaches have been proposed to date, no consensus exists on a particular algorithm as being the most appropriate. Traditional approaches have not been able to achieve consistent robust performance in terms of the commonly used performance measures such as detection rate, false alarm rate, and the average time to detection. With paradigms for automated integrated incident detection and response being addressed under the aegis of intelligent transportation systems, the need for a robust general incident detection approach is especially timely. The primary limitations of existing approaches are their inconsistent performance at the current site, lack of transferability to new sites without extensive off-line training and calibration, and nonadaptability to demand and supply changes in a traffic network. Abdulhai and Ritchie (1) provide a detailed discussion of the characteristics desirable in a universally applicable incident detection approach.

Over the past few decades, incident detection algorithms based on several different techniques have been developed, though only a few have been implemented in practice. One of the most commonly used algorithms is the California algorithm (2, 3) which uses the decision tree technique. It is based on the principle that an incident is likely to significantly increase vehicle occupancy upstream while reducing it downstream. Levin and Krause (4) proposed a Bayesian approach that incorporates historical information on incidents into a probabilistic model. The approach uses optimal thresholds obtained by maximizing the probability of a correct decision. This approach also aided in better calibrating the California algorithm.

The Time Series algorithm (5) employs statistical forecasting of traffic features using a time series model, to provide short term forecasts of the traffic parameters. Stephanedes and Chassiakos (6) proposed filtering the raw traffic data prior to its use in the incident detection
algorithm to reduce the number of false alarms generated by high-frequency random traffic fluctuations.

The McMaster algorithm (7, 8) is based on a two-dimensional analysis of traffic data. The algorithm demarcates a presumed flow-occupancy relationship into three or four areas corresponding to different traffic operating states. Incidents are detected by analyzing changes in traffic measures in a short time period, and tracking the transition across states. Ritchie and Cheu (9) use the classification capability of a neural network to devise algorithms for incident detection. The algorithms are calibrated through off-line tests and implemented using static parameters. Under fuzzy logic (10), current traffic conditions are converted into an index value to reflect the possibility of an incident.

Though the above algorithms have all been tested and perform satisfactorily under certain situations, they lack certain characteristics for universal application. All of them require calibration of the relevant parameters for a specific site. Also, they do not incorporate continuous learning and hence lack adaptability. Abdulhai and Ritchie (1) introduce a framework for a universally transferable freeway incident detection algorithm. It also incorporates measures such as incident likelihood probabilities, and the asymmetric costs associated with misclassification of a traffic situation, for developing an integrated detection and response methodology. Though prior probabilities can be estimated from previous data, the cost of misclassification is subjective and may result in erroneous training of the neural network.

The primary objective of this paper is to introduce a framework for incorporating a neural net based continuous learning capability to existing algorithms commonly used in practice, such as the California algorithm. Such a capability allows transferability, enhances performance with time in service, and ensures adaptability to demand and supply changes at the current site,
through dynamic training of the parameters based on on-line data. An added advantage of this approach is that existing traffic measures (for example, volume, speed, occupancy, etc.) and associated data used in those algorithms are sufficient, circumventing the need for new traffic measures, new threshold parameters, variables requiring subjective decisions, and/or explicit off-line calibration. Least squares (LS) and error back propagation (EBP) based neural network implementations are analyzed for the California and McMaster algorithms.

The paper has six sections. The next section provides a theoretical discussion of the LS technique and the EBP algorithm within a neural net framework. This is followed by a description of the implementations of the California and McMaster algorithms with and without neural networks. Experiments for analyzing the algorithms are then described. The results are discussed, followed by concluding comments.

**METHODOLOGY**

The LS technique and the EBP algorithm are popular neural network approaches (11) used for training algorithms. This section presents a conceptual description of these approaches.

**Feed-forward neural networks**

Feed-forward neural networks consist of one or more layers of non-linear processing elements or units where the output of each layer feeds the next layer of units (11). They may include one or more layers of hidden units between the input (which is not a neural computing layer) and the output layer. A feed-forward neural network can be viewed as a system transforming a set of input patterns into a set of output patterns. The parameters used in the transformation of the input patterns to the output patterns are called synaptic weights. In case of multi-layered neural networks, the units belonging to neighboring layers are connected by sets of synaptic weights. A neural network can be trained to provide a desired response (output) to a given input. It achieves
such a behavior by adapting its synaptic weights during the learning phase on the basis of some learning rules and past data. The output from the neural network obtained on the basis of the input and the present synaptic weights is compared with the corresponding actual output (desired output), and the synaptic weights are subsequently updated.

**Least Squares Technique**

The LS technique (11) trains the synaptic weights by minimizing the sum of the squares of the errors between the output from the neural network and the desired output that is available as part of the training set. The training set consists of sets of inputs and desired outputs. The neural network uses the input of the training set and provides output based on the existing synaptic weights. By minimizing the error between the output from the neural network and the desired output, the synaptic weights are updated and the neural network is trained for future use.

Consider the single-layered neural network in which a particular pattern of input $x_{j,k}$ ($j = 1, 2, ..., n_i$, where $n_i$ is the number of input variables required by the neural network; and $k = 1, 2, ..., m$, where $m$ corresponds to the number of input and output training data sets available for the neural network) to the neural network produces an analog output $\hat{y}_{i,k}$ consisting of the elements:

$$\hat{y}_{i,k} = \sigma\left( \sum_{j=0}^{n_i} w_{ij} x_{j,k} \right) \quad \forall \quad i = 1, 2, ..., n_0,$$

(1)

where $x_{0,k} = 1$; $n_0$ is the number of output variables obtained from the neural network and $w_{ij}$ are the synaptic weights relating the $j$th input to the $i$th output of the training set, and $\sigma$ is a continuous function.
For a network consisting of \( n_0 \) output units, the discrepancy between the desired outputs \( y_{i,k} \) and the estimates \( \hat{y}_{i,k} \) provided by the neural network can be measured by the quadratic error \( E_k \) given by:

\[
E_k = \frac{1}{2} \sum_{i=1}^{n_0} (y_{i,k} - \hat{y}_{i,k})^2 \quad \forall \quad k = 1, 2, \ldots, m.
\]

(2)

In the LS technique, the training of single-layered neural networks is based on the minimization of \( E \):

\[
E = \sum_{k=1}^{m} E_k = \frac{1}{2} \sum_{k=1}^{m} \sum_{i=1}^{n_0} (y_{i,k} - \hat{y}_{i,k})^2
\]

(3)

**Error Back Propagation Algorithm**

The EBP algorithm \((11)\) also trains the neural network by minimizing the squared errors. However, input and output sets from the training data are taken one at a time, unlike the LS technique where they are taken together. The output based on the existing synaptic weights and the inputs is compared with the corresponding desired output. The squared error between the desired output and the neural network output is minimized to update the synaptic weights. This is done for each set of inputs from the training set.

Hence, the EBP algorithm is derived by sequentially minimizing the objective function \( E_k \) defined earlier for \( k = 1, 2, \ldots, m \). Consider a network with one layer of hidden units (see Figure 3). Assuming that the pattern \( x_{j,k} \) is the input to the network, the corresponding outputs \( \hat{h}_{l,k} \) to the hidden units are:

\[
\hat{h}_{l,k} = p\left( \sum_{j=0}^{n_l} v_{lj} x_{j,k} \right) \quad \forall \quad l = 1, 2, \ldots, n_h,
\]

(4)
where \( x_{0,k} = 1 \); \( n_h \) is the number of units in the hidden layer, \( v_{lj} \) are the synaptic weights relating the \( j \)th input to the \( l \)th unit in the hidden layer and \( \rho \) is a continuous function. The outputs of the network are given by:

\[
\hat{y}_{i,k} = \sigma(\sum_{l=0}^{n_h} w_{il} \hat{h}_{l,k}) \quad \forall \quad i = 1, 2, ..., n_0, \tag{5}
\]

where \( h_{0,k} = 1 \); \( n_0 \) is the number of output variables obtained from the neural network and \( w_{il} \) are the synaptic weights relating the \( l \)th unit in the hidden layer to the \( i \)th output of the training set, and \( \sigma \) is a continuous function. The update equations for the synaptic weights \( w_{il} \) are obtained as follows:

\[
w_{il,k+1} = w_{il,k} + \alpha \varepsilon^0_{i,k} \hat{h}_{l,k} \tag{6}
\]

where \( \alpha \) is the learning rate. The learning rate tells the neural network how slowly to progress. The weights are updated by a fraction of the calculated error each time to prevent large fluctuations about the average values.

\[
\varepsilon^0_{i,k} = -\sigma'(\sum_{l=0}^{n_h} w_{il,k} \hat{h}_{l,k}) \frac{\partial \phi(e_{i,k})}{\partial \hat{y}_{i,k}} \tag{7}
\]

where \( e_{i,k} \) is an error function. The synaptic weights of the hidden layer are also upgraded in a similar manner. If the output of the network is binary, that is \( \sigma(.) = \tanh(.) \), and the network is trained by minimizing the quadratic error, \( \phi(e_{i,k}) = \frac{1}{2} e_{i,k}^2 \), the synaptic weights are updated using:

\[
\varepsilon^0_{i,k} = (1 - \hat{y}_{i,k}^2)(y_{i,k} - \hat{y}_{i,k}) \tag{8}
\]

**ALGORITHMIC IMPLEMENTATION**
This section describes the logic of the California and McMaster algorithms, and the corresponding continuous learning neural net implementations using the LS and EBP approaches.

**California Algorithm**

Four traffic measures, all based on occupancy data, DOCC, OCCDF, OCCDRF, DOCCTD, and the previous state, serve as input units to the California algorithm. The traffic measures are functions of occupancy data with respect to adjacent detector stations for successive time intervals (2).

Occupancy is obtained by averaging values over all instrumented lanes at a location on the freeway over a one minute interval. The algorithm is usually executed for each adjacent pair of detector stations on the freeway. The traffic measures are compared with threshold parameters that are calibrated off-line, in order to determine whether an incident occurred. This is done by estimating the current state of the system. As shown in Figure 1, we use the popular version 8 of the California algorithm, which is classified into the nine states, 0 through 9 (2). States 0 through 6 represent incident free states whereas the states 7 and 8 represent incident states.

The algorithm suppresses incident detection at a station for a period of 5 minutes after a compression wave is detected at the downstream detector station. While the algorithm uses the concept of persistence and shock wave detection capabilities to limit false alarms, the fixed threshold values are based on a trial and error procedure. Hence, these fixed threshold values may not provide good detection and false alarm rates under varying traffic patterns.

In our experiments, these static thresholds are calibrated using the proposed neural network methodology for specific network geometry using simulated data from the INTRAS
model for several demand levels, and incident types, ensuring a robust set of thresholds for the conventional algorithms.

*Neural Network Implementations of the California algorithm*

The LS and EBP algorithms are implemented to update the thresholds of the California algorithm. Unlike the conventional California algorithm which assumes constant thresholds after calibration, the neural network implementations continuously update the thresholds over time based on experience. This circumvents the need for off-line calibration for installation at a new site, especially if historical data is not available.

**Least Squares Technique Implementation** The LS technique implementation for the California algorithm is shown in Figure 2. The input has 5 units, which are the 5 inputs to the California algorithm as well. They feed their values to the units of the output layer where the decision making is done. The output layer has 12 units. Each of these take values greater than or less than 0, which are then matched to one of the 9 states defined in the algorithm. The thresholds are represented by the synaptic weights of some units of the output layer. For example, in Figure 2, the synaptic weights of output units 2, 4, 5, 11, and 12, represent the thresholds for the LS implementation of the California algorithm. The synaptic weights are updated by comparing the output layer unit values corresponding to the predicted state with the corresponding values for the actual state. The updating of the synaptic weights is done by considering several time intervals together (corresponding to the consideration of several sets of data simultaneously).

Let the input units $x_{1k}, x_{2k}, \ldots, x_{5k}$, corresponding to a time interval k constitute the matrix $X$, and the units of the output layer be $\hat{y}_{1k}, \hat{y}_{2k}, \ldots, \hat{y}_{12k}$. The values for output units corresponding to the actual states, $y_{1k}, y_{2k}, \ldots, y_{12k}$, constitute matrix $Y$. The thresholds are represented in a weight matrix $\|w_{ij}\|$ denoted by $W$. The updated $W$, obtained by minimizing the
sum of the squares of the errors between the desired output and the output from the neural network, is represented as:

\[ W = YX^T(XX^T)^{-1} \]  \hspace{1cm} (9)

**Error Back Propagation Algorithm Implementation** The EBP algorithm consists of 2 neuron layers including 1 hidden layer as shown in Figure 3. The input consists of 5 units. The output and hidden layers each consist of 12 units. The five input units feed their values to the hidden layer where the decision making is done. The output units take binary values, -1 or 1. The specific combination is matched with one of the 9 states discussed in the California algorithm. As seen in Figure 3, the thresholds are represented by the synaptic weights of some units of the hidden layer. The synaptic weights are updated each time interval by comparing the output layer unit values corresponding to the predicted state with those corresponding to the actual state at every time interval using traffic state data from the previous time interval, and current time interval traffic data.

Let the binary values for each output unit corresponding to the actual states be \( r_1, r_2, \ldots, r_{12} \).

Also, let the five thresholds \( T_1, T_2, \ldots, T_5 \) be represented by the synaptic weights \( w_1, w_2, \ldots, w_5 \).

The back propagation error \( \varepsilon_n \) corresponding to synaptic weight \( w_n \) of hidden unit \( n \) is given by:

\[ \varepsilon_n = (1 - \hat{y}_n^2)(r_n - \hat{y}_n), \quad n = 1, 2, \ldots, 12 \]  \hspace{1cm} (10)

The synaptic weights are obtained subsequently as follows:

\[ w_n = (w_n + \alpha \varepsilon_n) / (\alpha + 1), \]  \hspace{1cm} (11)

where \( \alpha \) is the learning rate (denotes the rate at which the synaptic weights are trained by the new pairs of predicted and actual output layer unit values). The learning rate is the weight given to the current data set in training the synaptic weights of a neural network as indicated in equation (11). It can vary from 0 to 1. A low value indicates slower adaptation to changes in the traffic pattern,
while a high value makes it very sensitive to changes leading to instability of the synaptic weights. In our experiments, the learning rate is assumed to be 2%, based on a sensitivity analysis of the data.

**McMaster Algorithm**

The McMaster algorithm (7) uses flow, occupancy, and speed (if available) from a single station to automatically detect congestion due to incidents near that station. It uses a catastrophe theory model to describe the relation between flow, occupancy, and speed. The logic, shown in Figure 4, classifies traffic conditions into four states using three parameters $O_{crit}$, $V_{crit}$, and $a$. Uncongested flow-occupancy operation occurs in State 1. An incident causes movement from state 1 to congested states 2 or 3.

The boundaries are calibrated off-line for each system on the basis of the traffic patterns resulting from several demand levels. The calibration is done by plotting the volume and occupancy values corresponding to each time interval of the historical data on a volume-occupancy graph. The points corresponding to incident conditions are highlighted. The boundaries are set so as to distribute the incident related points to states 2 and 3 and the non-incident related points to states 1 and 4.

**Neural Network Implementations of the McMaster algorithm**

The neural network implementations are based on the logic of the McMaster algorithm. However, unlike the McMaster algorithm, they do not assume fixed values for the parameters $O_{crit}$, $V_{crit}$ and $a$. An initial guess of these parameters is continuously updated according to actual data from the site using either the LS technique or the EBP algorithm. Volume and occupancy data are fed to the neural network and outputs are obtained based on the current
parameter values (synaptic weights). These outputs are compared with desired outputs, based on which the synaptic weights are updated for the next run of the neural network.

**Least Squares Technique Implementation** Volume and occupancy are used as inputs. Based on the current parameter values (Ocrit, Vcrit, and a), the output is determined. Akin to the LS implementation of the California algorithm, the sum of squared errors between the output from the neural network and the true output is minimized.

**Error Back Propagation Implementation** The EBP implementation uses the volume and occupancy data from the current time interval to determine their state using existing parameter values. This state is compared with the desired state to update the parameter values for the next time interval.

**EXPERIMENTS**

The two neural network based approaches are tested with simulated data obtained using the INTRAS simulation model (12). A 24 km section of the Borman Expressway (I-80/I-94) in northwestern Indiana is considered. A test link 2.4 km long is simulated with several detectors. Data for analyzing the algorithms is taken from detectors 01 and 02 spaced 0.8 km apart. This is the normal spacing of detectors on I-80/I-94.

Each experiment is simulated for a nine hour period consisting of several incident and non-incident periods of random lengths (except for experiments involving incidents of fixed lengths). The various experimental parameters are discussed hereafter:

**Network Geometry**

The number of lanes is varied between 2 and 3 on the test link. Also, the off-ramp from the test link is either closed or open.

**Demand**
In these experiments, uniform demand is considered. Three congestion levels: low, medium, and high, are analyzed. The performance of the algorithms under time-varying demand is discussed in Peeta and Das (13), and Das (14).

**Incident Type**

Various types of incidents are considered according to their duration and severity. The severity of the incident is indicated by one or two lane blockages. Incident durations vary from 5 to 35 minutes. In some experiments, incidents of specific duration are tested.

The incident detection algorithms are tested using the output of each experiment. The McMaster and California algorithms are implemented with (calibrated) fixed thresholds as discussed previously. Three performance measures are used for analyzing the effectiveness of incorporating a continuous learning capability: detection rate (DR), false alarm rate (FAR), and the average time to detection (TTD). They are calculated as follows:

- **DR** = \( \frac{\text{Number of incidents detected by algorithm}}{\text{Number of incidents simulated}} \times 100\% \)
- **FAR** = \( \frac{\text{Number of incident alarms in absence of real incidents}}{\text{Total number of incident alarms}} \times 100\% \)
- **TTD** = Time taken to detect an incident after its occurrence.

**OBJECTIVES OF EXPERIMENTS**

The experiments aim to test the performance of the conventional California and McMaster algorithms and their neural net implementations for varying geometry, demand levels, and types of incidents. Three sets of experiments are performed: (i) Experiment Set I evaluates the performance of the algorithms under varying demand levels. The experiments are performed for 2-lane and 3-lane networks; (ii) Experiment Set II addresses the performance of the algorithms under temporary ramp closures. These experiments are performed for high demand levels. Two
and three lane networks are considered; (iii) Experiment Set III evaluates the performance of the algorithms under different incident frequencies. The three scenarios considered are, one incident every 25 minutes, 40 minutes, or 60 minutes. In all the three sets of experiments, the incidents simulated are random in duration and severity (number of lanes blocked). Das (14) discusses additional experiments, including the performance of the neural network implementations under different types of incidents (in terms of severity and duration), and for incidents of specific durations.

**RESULTS**

The results emphasize the superior performance of the neural net implementations of the California and McMaster algorithms over the corresponding conventional ones. Figures 5 through 11 discuss some of the results.

Figure 5 compares the EBP implementation of the California algorithm for three different demand levels. The results indicate that eventually detection rates close to 100% are achieved under all three demand levels. The algorithm achieves a superior detection rate faster under medium demand and slower under high demand. This is because the effect of an incident on the traffic flow pattern is most emphatic when the demand is near capacity. Hence, the algorithm gets the best training sets in presence of medium demand leading to faster learning. Under high demand, congestion is already present making the occurrence of an incident less discernible. Similarly, when demand is low, an average incident does not affect the traffic flow pattern appreciably. Since the starting guesses are arbitrary, the detection rate starts from 0% and exhibits superior performance with time. For similar reasons, the false alarm rate decreases fastest under medium demand. The false alarm rate is always zero under low demand, while it reduces gradually in the presence of high demand.
Figure 6 compares the performance of the EBP and LS implementations of the California algorithm for medium (uniform) demand levels and a three lane scenario. It confirms the _a priori_ expectation that the EBP implementation leads to faster learning compared to the LS implementation as EBP involves sequential training of the data after each time interval. The conventional California algorithm exhibits near constant detection (90%) and false alarm (9%) rates which are inferior to the long-term detection (100%) and false alarm (0%) rates of the neural net implementations.

Figures 7 and 8 compare the performance of the EBP implementations of the California and McMaster algorithms with those of the conventional algorithms for medium demand. The EBP implementation of the McMaster algorithm exhibits faster learning with respect to the detection rate while the EBP implementation of the California algorithm exhibits faster learning with respect to the false alarm rate. The calibrated conventional California and McMaster algorithms exhibit near uniform performance over time. The faster learning of the McMaster algorithm with respect to detection rate is because it has a lesser number of thresholds to be trained (3 thresholds as opposed to 5 thresholds for the California algorithm). The superior performance of the California EBP implementation vis-à-vis the false alarm rate is because it suppresses incidents for 5 minutes after their occurrence, thereby being more conservative.

Figures 9 and 10 compare the effectiveness of the EBP implementations of the California and McMaster algorithms with the corresponding conventional ones for ramp closure under medium demand. The results emphatically illustrate the robustness of the neural net implementations. The conventional algorithms have a significant deterioration in their performance during the two periods of ramp closures. The neural net implementations, while
starting with arbitrary guesses for the thresholds, exhibit steady learning even during ramp closures. Thereby, they have a detection rate close to 100% during the second ramp closure.

Figure 11 compares the performance of the EBP implementations of the California algorithm for different incident frequencies for medium demand. It is observed that higher incident frequencies exhibit faster learning. This is because relatively more incident data sets are available for training in traffic networks with high incident frequencies.

**CONCLUSIONS**

Neural network based continuous learning implementations of the California and McMaster algorithms are developed. Two neural network approaches, Least Squares and Error Back Propagation, are used to develop the continuous learning versions of these algorithms using their existing logic and variables. Results suggest that continuously training the appropriate parameters online as opposed to off-line calibration enhances their performance with time in service, ensures automatic adaptation to any long-term changes in demand and/or supply, and allows transferability to new sites. The continuous learning feature circumvents the extensive off-line training and calibration requirements of current approaches (including neural nets), which are based on static threshold parameters. A key implication from the perspective of current practice is that existing variables, data, and threshold parameter variables suffice. Though both LS and EBP implementations enhance performance with time in service, the EBP based implementations exhibit better learning. Also, in both implementations, the McMaster algorithm performs better because it has fewer thresholds to be trained. The results suggest that incorporating currently used algorithms within a continuously trained neural network framework provides enhanced characteristics that can be exploited in traffic systems with advanced
technologies. The effectiveness of the neural network implementations is more emphatically highlighted under time-varying demand (13, 14).

REFERENCES


FIGURE 1 Decision Tree for the California Algorithm.
FIGURE 2 Least Squares Implementation of the California Algorithm.
FIGURE 3 Error Back Propagation Implementation of the California Algorithm.
FIGURE 4  Flow-occupancy Graph for the McMaster Algorithm.
FIGURE 5  EBP Implementation of the California Algorithm under Different Demand Levels.
FIGURE 6  EBP and LS Implementations of the California Algorithm.
FIGURE 7 EBP Implementations of the McMaster and California Algorithms.
FIGURE 8 EBP Implementations of the McMaster and California Algorithms.
FIGURE 9  EBP Implementations of the McMaster and California Algorithms under Ramp Closure.
FIGURE 10 EBP Implementations for the McMaster and California Algorithms under Ramp Closure.
FIGURE 11  EBP Implementation of California Algorithm under Different Frequencies of Incident Occurrence.