Project Number 288094

eCOMPASS

eCO-friendly urban Multi-modal route PlAnning Services for mobile uSers

STREP
Funded by EC, INFSO-G4(ICT for Transport) under FP7

eCOMPASS – TR – 039

Analysis of the State-of-the-Art for Traffic Incident Detection

T. Diamantopoulos

April 2013

Project Number 288094

eCOMPASS

eCO-friendly urban Multi-modal route PlAnning Services for mobile uSers

STREP

Funded by EC, INFSO-G4(ICT for Transport) under FP7

eCOMPASS – TR – 039

Analysis of the State-of-the-Art for Traffic Incident Detection

T. Diamantopoulos

April 2013

eCO-friendly urban Multi-modal route PlAnning Services for mobile uSers

FP7 - Information and Communication Technologies

Grant Agreement No: 288094
Collaborative Project
Project start: 1 November 2011, Duration: 36 months

Analysis of the State-of-the-Art for Traffic Incident Detection

Workpackage: WP2 - Algorithms for Vehicle Routing
Contributing Partners: CERTH

Nature: ☒ Report ☐ Prototype ☐ Demonstrator ☐ Other

Dissemination Level:
☒ PU: Public
☐ PP: Restricted to other programme participants (including the Commission Services)
☐ RE: Restricted to a group specified by the consortium (including the Commission Services)
☐ CO: Confidential, only for members of the consortium (including the Commission Services)

The eCOMPASS project (www.ecompass-project.eu) is funded by the European Commission, DG CONNECT (Communications Networks, Content and Technology Directorate General), Unit H.5 - Smart Cities & Sustainability, under the FP7 Programme

The eCOMPASS Consortium

Computer Technology Institute & Press 'Diophantus' (CTI) (coordinator), Greece

Centre for Research and Technology Hellas (CERTH), Greece

Eidgenössische Technische Hochschule Zürich (ETHZ), Switzerland

Karlsruher Institut fuer Technologie (KIT), Germany

TOMTOM INTERNATIONAL BV (TOMTOM), Netherlands

PTV PLANUNG TRANSPORT VERKEHR AG. (PTV), Germany

Contents

1 Introduction

Nowadays, improving traffic flow in road networks has grown to be a challenging problem that has major socio-economical and environmental effects. Thus, the notion of *Intelligent Transportation Systems (ITS)* has emerged as a fully organized solution to the task of alleviating traffic conditions. According to the Traffic Incident Management Handbook [1], studies have shown that traffic congestion at U.S. roadways costs more than 85 billion dollars per year. In addition, almost 25% of congestion is due to traffic incidents. An ITS could generally use various methods to improve traffic conditions. Traffic prediction and jam prediction approaches have been widely used during the last decade in order to alleviate congestion and improve routing of individuals or fleets. This review, however, focuses mainly on the difficult task of predicting traffic incidents. Thus, the rest of this review defines the problem and summarizes the main approaches towards its solution.

There are numerous incident detection systems, and many of them are actually used in real world. The basic functionality of an incident management system, though, is common. The flow chart of such a system is shown in Figure 1.

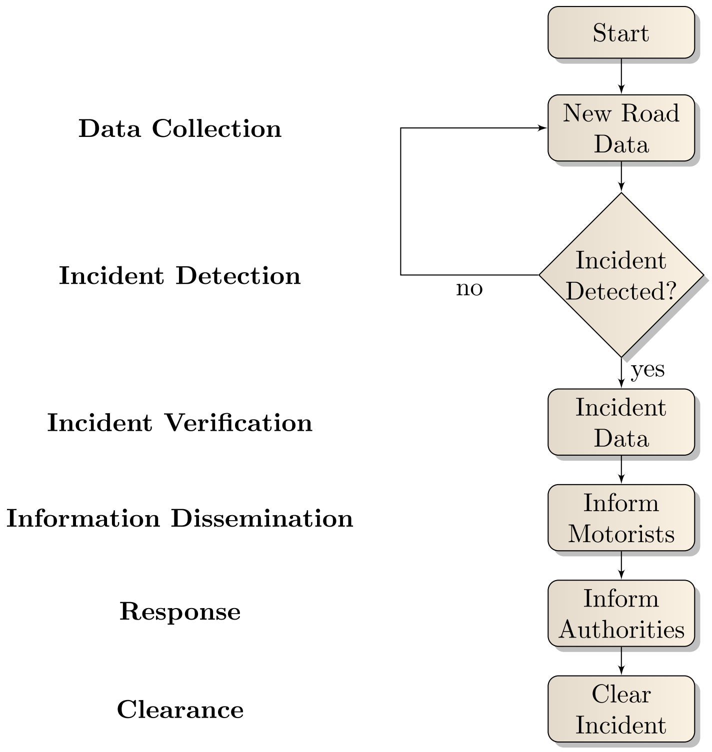

Figure 1: Basic Flow Chart of an Incident Management System

As shown in Figure 1, the system continuously collects new road data and checks whether any incidents have occurred. Upon detecting an incident, it is usually verified by a human observer who is responsible for gathering the data for the incident as it is given by the system. Incident data involves the location of the incident, the vehicles involved, possibly the lanes it affects, etc. Upon verification, the motorists have to be informed at once usually by signs on the road or even by local radio stations. This is vital to avoid chain incidents. After that, the authorities are informed in order to take over and clear the incident, by e.g. removing any participating vehicles and restore normal flow of the road.

1.1 The Problem of Detecting Traffic Incidents

Traffic incidents are defined as non-recurring roadway events which may result in congestion. Examples of such events may include accidents, unpredictable weather conditions, or even artist tours. Further analyzing the term *non-recurring*, one should note that periodic congestion, such as the morning peak where several people drive to work, are not included in the category of incidents. In other words, incidents are modeled as deviations of the normal flow, as it would be at a particular time of the day. Due to the aforementioned issues, incident detection is considered a quite hard problem; incidents cannot be predicted using conventional congestion techniques since no previous data for the incidents are available.

Further formulating the problem comes down to defining the methodology for detecting non-recurring incidents. In general, a traffic incident detection system receives traffic data as input and is trained along the lines of identifying unexpected congestion. Evaluating such a system requires having a database with known events. Subsection 1.2 summarizes the different sources of data used by the algorithms, while subsection 1.2 comments on the measures used to evaluate traffic incident detection techniques. The various techniques are analyzed in Section 2 which is the main section of this report.

1.2 Traffic Descriptors

Traffic descriptors refer to the different traffic data forms that are acquired using multiple means. Before analyzing current literature on incident prediction, the source and form of traffic data are enumerated since the various techniques are usually bound to some specific input [2–5]. Thus, the various traffic descriptors are:

- **Traffic Flow:** Also known as *traffic intensity* or *traffic volume*. It is measured using *loop detectors*, i.e. systems that detect the presence (or absence) of a vehicle where they are installed. Thus, traffic flow is measured in vehicles per hour.

- **Occupancy:** It measures the concentration of the vehicles in some specific area. Similarly to traffic flow, occupancy is also usually measured using loop detectors, only the measured quantity is actually the time percentage that each detector is “on”.

- **Time Mean Speed:** It is calculated as the mean speed of vehicles in a specific road segment and it is measured in kilometers per hour. Although instantaneous speed probes drawn from GPS devices on vehicles may be used, in practice the use of two loop detectors is sufficient for calculating the time mean speed of the segment between them.

- **Space Mean Speed:** It is calculated using speed probes drawn from GPS devices. Instantaneous vehicle speeds form a trajectory for every vehicle, thus instead of using its location (as in time mean speed), the mean of all vehicle speeds within the road segment is calculated. In practice, when speed probes are available, space mean speed is preferred to time mean speed since it is considered more accurate.

- **Traffic Density:** It is defined as the number of vehicles per distance unit for a specific road segment. It can be calculated using loop detectors by computing the number of vehicles traveling simultaneously along a road segment, i.e. between two detectors.

According to C. Xie and C. Parkany [2], the aforementioned descriptors are highly popular for most transportation management centers, while the authors also note that several systems may use more than one descriptor. Traffic flow, occupancy and time mean speed are actually present in almost all systems. Apart from the above descriptors, other rarely used metrics may include average *travel time* per road segment, or even congestion-related metrics such as the distance between subsequent vehicles, known as *headway*, or the *queue length*.

1.3 Performance Measures

Although current literature in incident detection is vast [2–5], evaluating the techniques is a rather common ground. The performance measures of the algorithms are based on the fact that a set of incidents is known. Given the actual incidents, the techniques are easily evaluated against them. The main performance measures are [2]:

- **Detection Rate (DR):** It is defined as the percentage of the number of detected incidents divided by the number of actual incidents. It is determined for any given time interval as:

$$DR = \frac{NumberOfDetectedIncidents}{NumberOfActualIncidents} \cdot 100\% \tag{1}$$

  Despite it being a simple measure, the detection rate is quite useful since it captures the recall of the system, or, as one could say, the true positives. A system with low DR fails to capture a significant number of incidents. Generally, satisfactory systems have DR values above 88.3% [6], although this is usually a context-specific measure.

- **False Alarm Rate (FAR):** It is a measure of the false positives of an algorithm. It is defined as the number of falsely identified incidents divided by the number of algorithm applications:

$$FAR = \frac{NumberOfFalseAlarms}{NumberOfDetectorInvocations} \cdot 100\% \tag{2}$$

  Achieving low FAR is crucial. Studies have shown that when FAR is below 1.8%, the alarm triggers easily enough to distract users and make them not trust the system [6]. Finally, as noted in [2], two more definitions of FAR are used, one defined as the number of false alarms divided by the total number of (correctly and incorrectly) detected incidents and one defined as the number of false alarms for a specific time period. Although the former seems to capture satisfactorily the precision of the system, the definition shown in equation (2) is the one used more often.

- **Time-To-Detection (TTD):** Apart from detecting incidents, it is crucial that detection is performed in a timely manner. Thus, TTD is defined as the time spent from the time of occurrence of the incident until the time of its detection. For an incident i it is computed as:

$$TTD(i) = t_{detect}(i) - t_{occur}(i) \tag{3}$$

  The average TTD metric is computed by averaging over all successfully detected incidents. Lower TTD values are generally preferred, although the metric cannot be used on its own since it contains no information about the success rate of the detection algorithm.

The three aforementioned metrics are widely used in current literature to evaluate quantitatively the performance of incident detection techniques. Apart from them, common evaluation techniques include drawing curves, such as the *Receiver Operating Characteristic (ROC)* curve, which is satisfactorily applicable for most scenarios since the output of the algorithms is binary (incident - no incident). A ROC curve is created by plotting the true positive rate versus the false positive rate at various threshold settings.

Concerning, however, incident detection scenarios, *Activity Monitor Operating Characteristic (AMOC)* curves are preferable [7] due to the rarity of incident occurrence. The curve is created by plotting the TTD versus the FAR for various thresholds. Finally, certain other metrics may be used as a combination of the three aforementioned metrics. For instance, Stephanedes et al. [3] plot a DR-FAR curve, as an alternative to the known ROC curve. The relationship between DR, FAR, and TTD, which is explored by K. N. Balke [5], is shown in Figure 2.

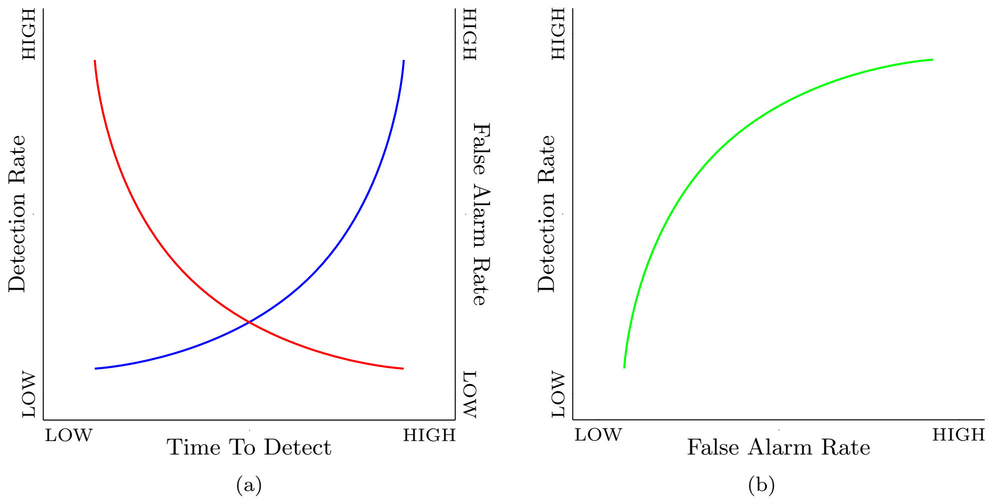

Figure 2: Relationship between Detection Rate, False Alarm Rate, and Time To Detect, as in [5]. Concerning diagram (a), the red line (■) is the False Alarm Rate versus the Time To Detect and the blue line (■) is the Detection Rate versus the Time To Detect. Concerning diagram (b), the green line (■) is the Detection Rate versus the False Alarm Rate.

As shown in Figure 2, the DR is qualitatively reversely proportional to the FAR. This is expected since attempting to detect more and more incidents results also in more false alarms. Concerning the TTD, when its maximum allowed threshold is increased, more incidents can be considered as detected either correctly or falsely, while reducing the threshold provides lower DR but also higher FAR.

1.4 Overview

Upon stating the problem and illustrating its significance, the main scope of this section was to identify the elements that are present in any ITS that practices incident detection. Thus, the form of the data was analyzed and the different approaches on measuring the performance of an incident management technique are clear. The rest of this report is organized as follows. Section 2 provides a taxonomy of current literature on traffic incident detection and summarizes the state-of-the-art approaches on the task at hand. Section 3 provides with real world implementations of detection techniques and comments on their impact. Finally, Section 4 concludes this report and provides useful insight for future research.

2 Traffic Incident Detection Techniques

Several researchers have tried to review current literature on traffic incident detection [2–5]. Although all reviews on the topic are very valuable, most of them are outdated since they only review algorithms that were implemented before 1990. This review aims to include both classic literature and current state-of-the-art on incident detection. Subsection 2.1 provides a taxonomy of traffic incident detection algorithms, while the 5 following subsections review different types of algorithms. This section is summarized in subsection 2.6 providing with main points for the current state-of-the-art in traffic incident detection.

2.1 Taxonomy of Traffic Incident Detection Techniques

Traffic incident detection has been a widely studied problem during the last few decades. Creating a taxonomy that covers all techniques is very difficult, if not impossible. However, classifying the techniques into different categories shall be useful in order to study their features, their similarities and differences. C. Xie and C. Parkany [2] classify the techniques into several categories, while including also non-automatic algorithms (i.e. driver-based) and distinguish them also according to the type of the roads (e.g. urban, arterial, etc.). Deviating from the aforementioned taxonomy, we classify the algorithms according to their applicability to different types of data. Figure 3 provides the taxonomy of the algorithms.

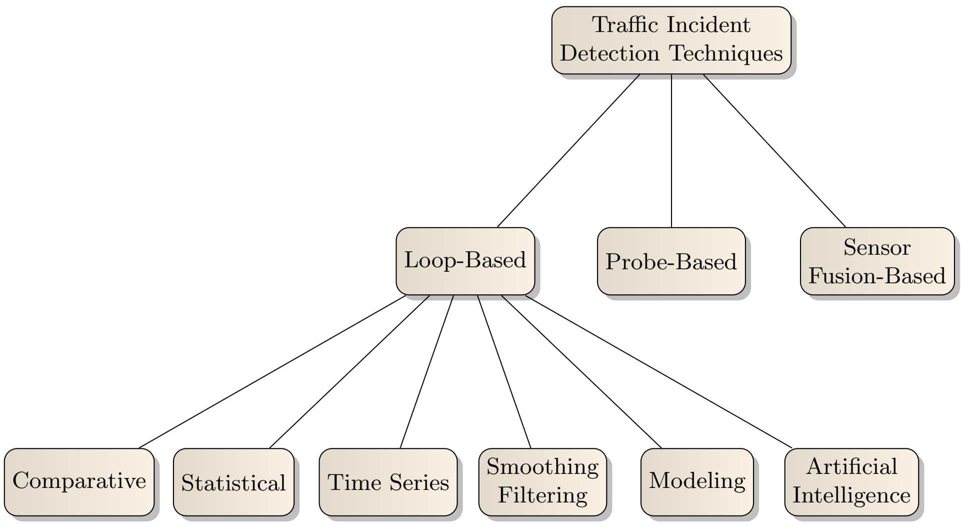

Figure 3: Taxonomy of traffic incident detection techniques.

As shown in Figure 3, there are five categories concerning the data used by the algorithms. *Loop-based* techniques, which are the most well-studied ones [2], use data from loop detectors, i.e. usually traffic flow or occupancy and less often time mean speed or traffic density (see subsection 1.2). The subcategories for loop-based algorithms follow the taxonomy of C. Xie and C. Parkany [2] which in turn originated from the one provided by K. N. Balke [5]. *Probe-based* systems rely on probe instantaneous speeds of GPS-equipped vehicles, thus the preferred descriptor is usually space mean speed. Finally, *sensor fusion-based* algorithms use multiple data sources, usually both loop detector data and probe data to detect incidents.

As already noted, several taxonomies may emerge by classifying algorithms according to their effectiveness in different scenarios, as in [2]. Concerning, however, this report, the main categories of algorithms are rather sufficient, since comments on the performance shall be integrated in the description of each algorithm. Subsections 2.2, 2.3, and 2.4 describe the algorithms of the loop-based, probe-based and sensor fusion-based categories respectively, while subsection 2.5 comments on omitted algorithms that do not match any of the above categories.

2.2 Loop-Based Techniques

Loop-based techniques are the most widely studied category of techniques due mainly to the fact that equipping vehicles with GPS locators has been difficult and/or costly until before the last decade of the past century. Thus, most incident detection systems acquired data from loop detectors, and the main measures used are flow and occupancy. This subsection provides an overview of different loop-based algorithms that are known for their performance. Each subcategory of algorithms, as shown in Figure 3, is analyzed in the following paragraphs.

2.2.1 Comparative Techniques

The comparative incident detection algorithms are based on comparing the value of a descriptor to specific threshold values. Probably the most widely known comparative algorithm is the *California algorithm* [8, 9], also known as the *Traffic Services Corporation (TSC) algorithm.* The California algorithm, in its simplest form, poses simple heuristic thresholds on traffic occupancy derived between two consecutive loop detectors. Given the detectors are A and B, the algorithm has three parameters:

- The absolute difference between the occupancy values of the two detectors:

$$P_1 = |OCC_A - OCC_B| \tag{4}$$

- The ratio of the difference of the occupancy values of the detectors divided by the occupancy value of the first detector:

$$P_2 = \frac{OCC_A - OCC_B}{OCC_A} \tag{5}$$

- The ratio of the difference of the occupancy values of the detectors divided by the occupancy value of the second detector:

$$P_3 = \frac{OCC_A - OCC_B}{OCC_B} \tag{6}$$

The three parameters P_1, P_2, and P_3 are compared to the thresholds T_1, T_2, and T_3 respectively. When all parameter values exceed the corresponding thresholds, then an incident is indicated by the algorithm. Although the California algorithm is generally effective, it is very simplistic and selecting appropriate values for the three thresholds is usually a difficult task. Consider also that the thresholds may differ among different loop detector pairs (e.g. an uphill and a downhill may differ significantly).

Upon further research [10], H. J. Payne and S. C. Tignor developed 10 different algorithms based on the California algorithm, known in current literature as *California #1*, *California #2*, etc. Out of all algorithms, *California #7* and *California #8* are the ones which had the best performance. California #7 is an attempt to reduce false alarms due to *compression waves.* The compression waves problem occurs when a sudden increase in occupancy is observer in both loop detectors that the California algorithm is applied. Thus, the difference of occupancy values equation (6) is replaced by the current occupancy in the first detector (OCC_A) to regard for recurrent high occupancy values. The other variation, California #8, is the most complex one with many parameters. It involves detection compression waves by categorizing traffic state into 9 different states and uses 5 threshold values to determine the current state.

The *All Purpose Incident Detection (APID)* algorithm was designed by P. H. Masters et al. [11] as a combination of the variations of the California algorithms. The main intuition of the APID algorithm lies in using different policies on different traffic conditions. Thus, the algorithm has three policies corresponding to low, medium and high traffic occupancy conditions, as well as two policies that take dynamic events into account, the first checks for compression waves while the second checks whether the conditions are persistent for a specific time interval.

Finally, another interesting algorithm is the *Pattern Recognition (PATREG)* algorithm, developed by J. F. Collins et al. [12], along with HIOCC (see subsection 2.2.3) in order to operate as a combined system. The PATREG algorithm uses traffic flow measurements to initially estimate travel times between subsequent loop detectors and subsequently construct the time mean speed for the corresponding road segment. This measure is then compared to the mean speed of the road at a given time. Although interesting, this approach is rather outdated and is not used today.

2.2.2 Statistical Techniques

As the name implies, statistical algorithms use statistical metrics to detect non-recurring traffic events. The first algorithm of this category is the *Standard Normal Deviate (SND)* algorithm, developed by C. L. Dudek et al. [13]. The algorithm uses historic data to determine the values of the mean and the standard deviation of the occupancy at different time intervals. The SND is defined as the number of deviations that the occupancy is away from the mean occupancy. Intuitively, the SND is a metric of the deviating behavior of traffic for a specific time interval. Using certain heuristic thresholds, any SND occupancy value exceeding these thresholds triggers the incident alarm.

Another significant statistical techniques is the *Bayesian* algorithm, proposed by M. Levin and G. M. Krause [14]. The algorithm uses the relative difference of occupancy values of loop detectors similarly to the California algorithm (see equations (5),(6)). However, instead of using predefined thresholds, exceeding values are determined using the conditional probability that the difference in occupancy between the detectors is caused by an incident. Since that probability is computed using the well-known Bayes theorem, the name of the algorithm is justified. Computing the probability is not trivial; it requires occupancy and traffic volume values in both incident and normal-flow conditions to construct the incident occurring distributions. Furthermore, an archive of occurring incidents is necessary to keep track of historical probabilities. Generally, the probability of an incident occurring in a t-minute time interval is given by the following equation [5]:

$$P_{INCIDENT} = \frac{\bar{N}_{INCIDENTS}}{N_{DETECTORS} \cdot t} \tag{7}$$

where $\bar{N}_{INCIDENTS}$ is the average number of incidents for the area of interest and $N_{DETECTORS}$ is the total number of loop detectors.

2.2.3 Time Series Techniques

The techniques of this category employ algorithms drawn from *Time Series Analysis*, assuming that traffic follows particular patterns. Probably the most well known model of this category is the *Auto-Regressive Integrated Moving Average (ARIMA)*, which was first introduced by Box and Jenkins [15]. The generic *Auto-Regressive Moving Average (ARMA)* model comprises of two components [16]:

- The *Auto-Regressive (AR)* part provides the current value X_t as the linear aggregate of p previous values:

$$X_t = \sum_{k=1}^{p} \phi_k X_{t-k} + \epsilon_t \tag{8}$$

  where ϵ_t is the error term and follows a Gaussian distribution of type $(0, \sigma_\epsilon^2)$ (White Noise).

- The *Moving Average (MA)* part provides the current value X_t as the aggregate of q previous error terms:

$$X_t = \sum_{k=1}^{q} \theta_k \epsilon_{t-k} + \epsilon_t \tag{9}$$

Hence, according to (8) and (9), the ARMA(p, q) model is given by:

$$X_t = \sum_{k=1}^{p} \phi_k X_{t-k} + \sum_{k=1}^{q} \theta_k \epsilon_{t-k} + \epsilon_t \tag{10}$$

or equivalently:

$$1 - \sum_{k=1}^{p} (\phi_k B^k) X_t = 1 + \sum_{k=1}^{q} (\theta_k B^k) \epsilon_t \tag{11}$$

where B is the *backwards shift* operator ($B^k X_t = X_{t-k}$). Upon differencing the series at the d-th degree ($(1 - B)^d X_t$):

$$1 - \sum_{k=1}^{p} (\phi_k B^k)(1 - B)^d X_t = 1 + \sum_{k=1}^{q} (\theta_k B^k) \epsilon_t \tag{12}$$

Finally, (12) describes an ARIMA(p, d, q) model.

In traffic incident detection, the ARIMA model was studied extensively by M. S. Ahmed and A. R. Cook, [17–19]. The authors employed an ARIMA$(0, 1, 3)$ model providing it with occupancy values for specific time intervals. Thus, given occupancy for the past $q = 3$ time intervals, the model provides an estimation of the occupancy value for the current time interval of the series. Given this value as well as the observed value for 1 interval before, the difference in occupancy between the current interval t and $t - 1$ can be computed. This difference can then be checked against specific thresholds to determine the existence of an incident. Finally, Note that although ARIMA has initially been adapted to occupancy measuring systems, other descriptors, such as speed or travel time, can easily be used in conjunction with this universal model, given in equation (12).

Another popular time series algorithm is the *High Occupancy (HIOCC)* algorithm, created by J. F. Collins et al. [12], as part of an ensemble technique with the PATREG algorithm (see subsection 2.2.1). The time series are modeled individually given the mean occupancy of each loop detector, which is computed for small time intervals. The algorithm attempts to detect consecutive high occupancy values (hence its name) and triggers an alarm when they exceed a specific threshold. Since high occupancy values on a detector may not be exactly “consecutive”, small gaps are filled when their duration is below a given threshold to avoid triggering multiple alarms for a single incident.

2.2.4 Smoothing or Filtering Techniques

Traffic data usually forms patterns, i.e. descriptor values that keep reappearing with certain trend and periodicity. Morning peaks, for example, are quite common causes for increased occupancy, yet they cannot be classified as incident-related congestion states. As opposed to incidents, these traffic states are recurrent. Smoothing or Filtering techniques aim to remove such data in order to reduce the FAR of the algorithm and isolate the non-recurrent incidents.

The first algorithm of this category is the *Double Exponential Smoothing (DES)* algorithm utilized by A. R. Cook and D. E. Cleveland [20]. In its simple form, *exponential smoothing* is a method of estimating the value of a series $\hat{x}_t$ as a weighted average of the previously estimated value $\hat{x}_{t-1}$ and the previously observed value x_{t-1}:

$$\hat{x}_t = \alpha \cdot x_{t-1} + (1 - \alpha) \cdot \hat{x}_{t-1} \tag{13}$$

where α is the *smoothing factor* and determines the weighting given to the real and the estimated values. The initial value of $\hat{x}_t$ is $\hat{x}_1 = x_0$. Although simplistic, exponential smoothing can generally be quite strong at estimating consecutive time series values. However, when the series has a specific trend, the algorithm is rather weak. To address this, one may use the DES, which introduces also a series b for the best estimate at all times, thus resulting in the following equations:

$$\hat{x}_t = \alpha \cdot x_t + (1 - \alpha) \cdot \hat{x}_{t-1} \tag{14}$$

$$b_t = \beta \cdot \hat{x}_t + (1 - \beta) \cdot b_{t-1} \tag{15}$$

where β is the *trend smoothing factor*. The initial value of $\hat{x}_t$ is $\hat{x}_1 = x_1$ and the initial value of b_t is $b_1 = x_1 - x_0$. Hence, A. R. Cook and D. E. Cleveland [20] utilized the model given by equations (14),(15) providing with different descriptors and concluding that traffic flow and occupancy yielded the most satisfactory results. Upon applying the algorithm, one can easily determine the difference between the real and estimated value $\hat{x}_t - x_t$. When this value deviates from zero it can indicate a possible incident. Thus, in practice, an upper threshold is set and when the difference exceeds it, an alarm is triggered.

A different set of smoothing algorithms comprise of the *Low-Pass Filter (LPF)* algorithms, also known as the *Detection Logic with Smoothing (DELOS) algorithms*, which were extensively studied by Y. J. Stephanedes and A. P. Chassiakos [21–23]. Given the total number of filtered intervals N, the equation used to remove high frequencies (i.e. noise in traffic data) is [5]:

$$x'_t = \sum_{k}^{M} \frac{x_{t-k}}{M+1} \tag{16}$$

where x'_t is the smoothed counterpart of the occupancy value x_t. The algorithms compare occupancy levels on two consecutive loop detectors. In its initial form, two occupancy values are considered for each detector, given equation (16), for three and five minute intervals respectively [21]. Using two values ensures lower FAR. Variations include using median values [22] or using exponential smoothing [23].

A similar line of work by A. Samant and H. Adeli [24] includes using *Discrete Wavelet Transform (DWT)* and *Linear Discriminant Analysis (LDA)*, forming thus the *DWT-LDA* method, in order to smooth the data. DWT is a transform of the input signal that

TODO: Write about DWT and LDA

2.2.5 Modeling Techniques

Modeling algorithms..

TODO: Write about Traffic Modeling Algorithms

2.2.6 Artificial Intelligence Techniques

Artificial Intelligence (AI) is certainly a quite broad field. In its general definition, AI may include virtually all *intelligent* decision-making algorithms. In the context of incident detection, however, the algorithms classified in this category are complex or "black box" algorithms [2]. Early research on incident detection employed *Fuzzy Logic (FL)* and *Artificial Neural Networks (ANNs)*, while lately the problem is also faced using other methods, such as *Decision Trees* or *Support Vector Machines (SVMs)*.

Artificial Neural Network Techniques ANNs have several variations which are used for a wide range of problems. Concerning the problem at hand, two ANN structures are preferred by current literature: *Multi-layer Feed-Forward Neural Networks (MLF-NNs)* and *Probabilistic Neural Networks (PNNs)*. An example NN is shown in Figure 4. Each node of the ANN of Figure 4 is called a *perceptron*. The perceptron is actually a processing unit, receiving a particular input and providing an output according to an *activation function*. Several activation are defined for

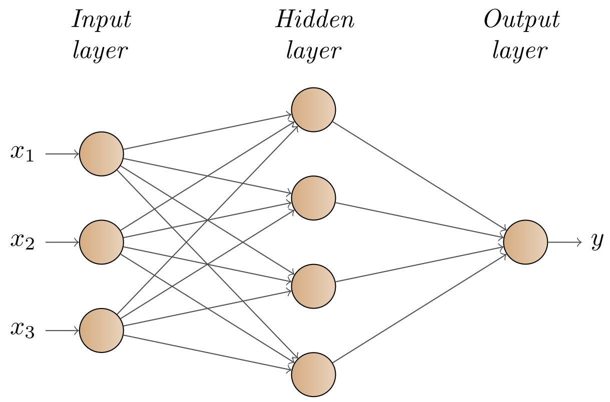

Figure 4: An Artificial Neural Network consisting of three layers

classification and regression problems, however they are omitted since they lie outsize the main scope of this report. The layers of an ANN are proportional to the complexity of the problem, i.e. the number of defined parameters. Thus, a network can be called Multi-layer as long as it has more than one layers. Although ANNs can have multiple different structures, one of the most well known ones is the structure shown in Figure 4, where the information moves only towards one direction: from the input to the output layer (with as may hidden layers as required). Hence, this is actually a Feed-forward network. Finally, each connection of an ANN carries a specific weight, which denotes the impact of the corresponding perceptron on the output. The process of training the network usually comes down to defining the weights, since the activation function is fixed. Out of the possible weight determining methodologies, the most widely used is *backpropagation*. The method includes two phases: propagation and weight update. During the former, the ANN is fed with input and provides with the outputs, while during the latter, the weights are updated according to the error between the known and the estimated output values.

MLF-NNs have been used by several researchers [25–29] in the field of traffic incident detection. The input layer of the ANNs can be provided with any measure, i.e. occupancy, speed, volume, etc., and the output is binary denoting whether there is an incident or not. As C. Xie and C. Parkany [2] note, correctly identifying incidents is a matter of properly training the network to most possible conditions. Cautious training is crucial in order to discriminate recurring and non-recurring congestion conditions, and thus to avoid false alarms.

Another widely used class of ANN techniques for incident detection are the ones based on PNNs [30–32]. The main discriminating feature of PNNs is that they have a *pattern layer* and a *summation layer* in place of the hidden layer. The input layer works as a distributor of the input to the perceptrons of the pattern layer, while the latter represent incident and free flow conditions. The summation layer concentrates all information into two perceptrons, one for each state as previously mentioned. Thus, upon normalizing, the perceptron of the output layer determines whether there is an incident or not. Although PNNs are quite adaptive to numerous traffic applications, MLF-NNs perform better in terms of higher DR and lower FAR.

Finally, a slightly different approach includes using DWT (see subsection 2.2.4) to improve the ANN implementations. Indicatively, reference must be given R. Prasenjit and B. Abdulhai [33] for using DWTs to train the PNN. Another interesting line of work is the one by A. Karim and H. Adeli [34, 35]. The authors form an input signal using both occupancy and speed values, and perform DWT in order to smooth the signal by discarding traffic fluctuations (see subsection 2.2.4). The data is then given to a *Radial Basis Function Neural Network (RBFNN)*, i.e. an ANN which has a *Radial Basis Function (RBF)* as its activation function. Upon further research [36, 37], the method is further improved using wavelet theory and LPFs to further smooth the time series before giving them as input to the RBFNN. Finally, A. Samant and H. Adeli [38, 39] propose the use of ACGNN

TODO: write about the ACGNN

As shown above, ANNs have been widely studied for the field of traffic incident detection. These models are also quite popular during the last decade, where several algorithms have emerged that improve on known ANN techniques. Indicatively, X. Cheng et al. [40] perform smoothing using wavelets before providing a back propagation ANN with the data, while D. Srinivasan et al. [41] propose a *Reduced Multivariate (RM)* polynomial neural network in order to handle the high dimensionality of the problem. Finally, an interesting evaluation of multiple different ANN implementations, including an MLF-NN and a PNN, is performed by D. Srinivasan et al. [42, 43]. The authors propose the use of a a *Constructive Probabilistic Neural Network (CPNN)* which consists of Gaussian components in the pattern layer.

TODO: Write about other Neural Network techniques, if any

Fuzzy Logic Techniques The theory of FL is based on the construction of certain rules which are applied to imprecise data. Intuitively, concerning traffic incident detection, FL techniques attempt to pose loose rather than strict thresholds to incident and non-incident situations. Thus, their output is actually a probability for an occurring incident. Finally, as C. Xie and C. Parkany [2] note, FL techniques perform quite satisfactorily for missing or inexact data, which is common in traffic scenarios.

FL-based techniques for traffic incident detection have been quite popular in early years [44–46]. The generic procedure of these techniques includes the following steps:

- **Data Extraction:** Refers to extracting the traffic data from the descriptors and transforming them to convenient form for the rest of the algorithm. This step may be different for each algorithm.

- **Data Fuzzification:** As its name implies, this step refers to fuzzifying the data, i.e. transforming crisp values (e.g. 80km/h) to membership functions (e.g. 70% high speed). Initially, the data crisps have to be identified in order to decide upon the definition of the fuzzy sets. After that, the fuzzification of the data is a straightforward mapping procedure from the crisps to the sets.

- **Fuzzy Rules Construction:** It refers to the construction of the rules that determine the output given the input. Given particular measure values, the combination of different rules outputs the probability of an occurring incident.

Recent approaches in FL use also the above principles. Interesting lines of research include using either simulated [47–49] or real data [50]. Indicatively, the proposed algorithm by Y. E. Hawas [48] seems quite satisfactory for discriminating among different incident types. In terms of effectiveness, the evaluation of S. Xiaorong et al. [50] is quite reliable, since the authors rely on real data. Their fuzzy learning classifier is proven to perform better than an ANN in both FAR and TTD.

Another well-studied class of algorithms for incident detection comprises of hybrid neuro-fuzzy approaches. Research on the subject initially focused on constructing the membership functions of the FL algorithm using ANNs. Indicatively, the approach by C. Hsiao et al. [51] involves neural training mechanisms to determine the threshold values of the FL rules. Similar approaches include using multiple sources of data, as in [52], while M. Viswanathan et al. [53] further develop the neuro-fuzzy method and compare it also to a newly proposed model of theirs, which extracts the rules using data mining techniques. On a different line of work, S. S. Ishak and H. M. Al-Deek [54, 55] propose the use of an FL-ANN deriving from the *Adaptive Resonance Theory (ART)*. The *Fuzzy ART* algorithm, as they call it, has a structure similar to an ANN. It initially clusters the different patterns and detection is thereafter performed on the corresponding cluster. A similar clustering approach to the problem is proposed by D. Srinivasan et al. [56]. The authors use *Least Squares Regression (LSR)* to train their model and evaluate it against a real dataset.

Support Vector Machines Techniques SVMs have been widely used in literature for various classification tasks. The main idea is to construct a hyperplane that sets apart the classes of a sample. A classification example is shown in Figure 5.

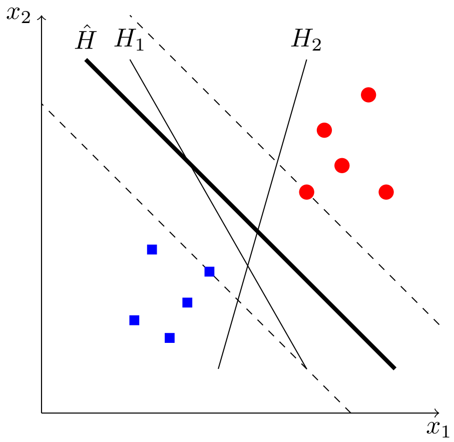

Figure 5: A Support Vector Machine, separating blue (■) from red (●) data points. H_1 and H_2 are two valid hyperplanes, and $\hat{H}$ is the optimal hyperplane (dashed lines denote maximum distances).

The example of Figure 5 concerns a two-dimensional space (with dimensions x_1 and x_2), i.e. the data instances are classified according to two attributes. For two dimensions, the hyperplanes are reduced to single lines. Although there may be various hyperplanes that separate a dataset, SVMs attempt to approximate the optimal hyperplane, i.e. the hyperplane that has maximum distance from instances on both sides. Maximizing the margin comes down to solving the *Quadratic Programming (QP)* problem. Current state-of-the-art suggests solving the problem using *Sequential Minimal Optimization (SMO)*.

F. Yuan and R. L. Cheu [57] were among the first researchers to address the problem of traffic incident detection using SVMs. The authors employ different non-linear kernels and evaluate the SVM implementations in both a simulated and a real dataset. Their results are encouraging, achieving higher DR and lower FAR when compared to an MLF-NN and a PNN. However, since selecting an appropriate kernel for the SVMs is usually a non-automated and non-trivial procedure, applying an SVM in a real scenario is rather limited. S. Chen et al. [58] attempted to address the problem by constructing an SVM *ensemble*. The ensemble is a system consisting of multiple SVMs with different kernels, while its output is determined using a combining scheme over the outputs of each individual classifier. The authors review different voting schemes and suggest new ones. The ensembles perform quite satisfactorily with respect to each individual SVM. Similar research on SVM ensembles was conducted by J. Xiao et al. [59, 60]. The authors also propose using a *Multiple Kernel Learning SVM (MKL-SVM)*, i.e. an ensemble of kernels (instead of SVM classifiers). The ensemble of MKL-SVMs, which is constructed, is quite effective compared to plain SVM ensembles, while it requires fewer classifiers.

Finally, recent research has also focused on improving the performance of SVMs using hybrid methods. T. Šingliar and M. Hauskrecht [7] improve the TTD of the classifier by realigning the output (detected incidents) with the data. To this end, the authors construct a dynamic Bayesian network which proves quite effective; the performance of the SVMs is satisfactory. Another interesting approach on improving SVMs is proposed by D. Zeng et al. [61]. The authors place SVMs in different loop detectors and use the *D-S evidence theory* algorithm to rate the confidence of each classifier. Their methods are proven to be effective against other implementations including an MLF-NN.

Other Artificial Intelligence Techniques

Trees, LSR, etc.

TODO: Write about the AI techniques

2.3 Probe-Based Techniques

The algorithms analyzed so far used data from loop detectors as their input. As shown, loop detectors have been widely deployed during the last few decades. Recent advances in technology have shown the way to acquiring more precise probe data using GPS locators or similar systems. Probe data can reveal incidents in road segments that would be covered more sparsely between loop detectors. Note, however, that probes provide with sample data from a small number of equipped vehicles, and this number may reflect quite a small percentage of the vehicle population (usually no more than 0.1% [2]). Consequently, an important metric that impacts the performance of any algorithm is the *penetration rate* of sensor-equipped vehicles.

Possibly the first approach in probe-based traffic incident detection is the *Managing Incidents and Traffic (MIT)* algorithm by E. Parkany and D. Bernstein [62]. The authors propose a fine-grained method, scanning for headways and even lane switches. Given the different travel times and volumes per lane, one could determine incidents as large deviations from lane to lane in the same road. Although the approach seems quite interesting, it is difficult to apply in a real world scenario. Indicatively, simulated probes are assumed to cover 50% of all vehicle data, which is not the case in most cases.

V. Sethi et al. [63] have implemented another well-known approach based on travel time and average speed per road segment. Their algorithm compares historical average values with current values in order to determine the occurrence of incidents. In the same context, M. W. Sermons and F. S. Koppelman [64] used multiple-form data ranging from travel time and position to running time derived from position. Although the granularity of every measure is different, the authors demonstrate the effectiveness of their approach by using detailed data. As one may note, the measures derived from proved can be either observed or constructed. An example of a constructed measures approach is the *Berkeley* algorithm by K. F. Petty et al. [65]. Instead of using average speed, the authors construct vehicle acceleration per road segment and propose using two different thresholds for speed and acceleration. Thus, when the probes indicate abrupt acceleration in otherwise free flow speed conditions, it is interpreted as an incident.

A different line of work, which is similar with the statistical loop techniques (see subsection 2.2.2), is proposed by K. N. Balke [5]. The author implemented an algorithm for the *Texas Transportation Institute (TTI)*, which uses thresholds similar to those of the SND algorithm (see subsection 2.2.2) to detect large deviations of travel time per road segment. However, the performance of the algorithm is rather unsatisfactory due to the low penetration rate of the data [2]. Other statistical algorithms include the *TRANSCOM's System for Managing Incidents and Traffic (TRANSMIT)* by K. C. Mouskos et al. [66, 67]. The system poses a threshold in travel time, given historical travel times and their deviations. Both TTI and TRANSMIT algorithms assume normal distributions over the data. Finally, similar research on the field has been conducted by B. Hellinga and G. Knapp [68]. The authors, however, assume log-normally distributed travel times and pose corresponding thresholds to both travel time and speed values.

As a final remark, although literature on probe-based detection seems rather limited, note that there are several loop-based algorithms that can easily be adapted to the probe scenario. For the applicability of the algorithms in different data scenarios, see subsection 2.6 of this report.

2.4 Sensor Fusion-Based Techniques

2.5 Other Techniques

Loop-Based Image Processing Techniques

Driver-Based Techniques
Arterial-Applicable Techniques

2.6 Summary

Having already analyzed several algorithms for traffic incident detection, one could wonder whether these algorithms are applicable to different scenarios. Generally, most algorithms, especially simple statistical and AI implementations, are easily adaptable to the problem at hand. Concerning traffic incident detection, the techniques analyzed are summarized in Table 1. Apart from a conclusive summary of this section, Table 1 contains applicability information for each algorithm concerning the type of data required.

Table 1: Traffic Incident Detection Algorithms and Supported Descriptors

Algorithm	Authors	Year	Applicability		
			Flow	**Volume**	**Speed**
Comparative					
California	Payne et al. [8, 9]	1976	✓		
California #7	Payne and Tignor [10]	1978	✓		
California #8	Payne and Tignor [10]	1978	✓		
APID	Masters et al. [11]	1991	✓	✓	
PATREG	Collins et al. [12]	1979	✓		
Statistical					
SND	Dudek et al. [13]	1974	✓	✓	
Bayesian	Levin and Krause [14]	1978	✓		
Time Series					
ARIMA	Ahmed and Cook [17–19]	1977	✓	✓	✓
HIOCC	Collins et al. [12]	1979	✓		
Smoothing/Filtering					
DES	Cook and Cleveland [20]	1974	✓	✓	✓
LPF	Stephanedes et al. [21–23]	1993	✓	✓	✓
DWT-LDA	Samant and Adeli [24]	2000	✓	✓	✓
Modeling					
...	...	...	...	...	...
...	...	...	...	...	...
Artificial Intelligence					
Neural Networks					
MLF-NN	Cheu et al. [25–27]	1991	✓	✓	✓
	Stephanedes and Liu [28]	1995	✓	✓	✓
	Dia and Rose [29]	1997	✓	✓	✓
PNN	Abdulhai and Ritchie [30, 31]	1999	✓	✓	✓
	Jin et al. [32]	2002	✓	✓	✓
	Srinivasan et al. [42, 43]	2004	✓	✓	✓
DWT-NN	Roy and Abdulhai [33]	2003	✓	✓	✓
	Adeli and Karim [34–37]	2000	✓	✓	✓

Algorithm	Authors	Year	Applicability		
			Flow	Volume	Speed
	Adeli and Samant [38, 39]	2000	✓	✓	✓
Other NN	Cheng et al. [40]	2010	✓	✓	✓
	Srinivasan et al. [41]	2008	✓	✓	✓
Fuzzy Logic					
FL	Chang [44, 45]	1994	✓	✓	✓
	Lin and Chang [46]	1998	✓	✓	✓
	Yaguang and Anke [47]	2006	✓	✓	✓
	Xiaorong et al. [50]	2007	✓	✓	✓
	Hawas [48]	2007	✓	✓	✓
	Binglei et al. [49]	2008	✓	✓	✓
Neuro-Fuzzy	Hsiao et al. [51]	1994	✓	✓	✓
	Lee et al. [52]	2005	✓	✓	✓
	Viswanathan et al. [53]	2006	✓	✓	✓
Neuro-FL clusters	Ishak and Al-Deek [54, 55]	1998	✓	✓	✓
	Srinivasan et al. [56]	2006	✓	✓	✓
SVM					
SVM	Yuan and Cheu [57]	2003	✓	✓	✓
SVM ensemble	Chen et al. [58]	2009	✓	✓	✓
	Xiao and Liu [59, 60]	2012	✓	✓	✓
Other SVM	Šingliar and Hauskrecht [7]	2007	✓	✓	✓
	Zeng et al. [61]	2008	✓	✓	✓

Thus, Table 1

3 Real World Implementations

4 Conclusion

References

[1] Federal Highway Administration. Traffic incident management handbook. Technical report, U.S. Department of Transportation, 2010.

[2] Xie C. Parkany E. A complete review of incident detection algorithms & their deployment: What works and what doesn't. Technical report, The New England Transportation Consortium, NETCR37 Project No. 00-7, February 7, 2005, 2005.

[3] Y. J. Stephanedes, A. P. Chassiakos, and P. G. Michalopoulos. Comparative performance evaluation of incident detection algorithms. *Transportation Research Record: Journal of the Transportation Research Board*, 1360:50–57, 1992.

[4] Hani S. Mahmassani and J. Peterman. *Evaluation of incident detection methodologies.* Center for Transportation Research, Bureau of Engineering Research, the University of Texas at Austin, 1999.

[5] Kevin N. Balke. *An evaluation of existing incident detection algorithms.* Texas Transportation Institute, the Texas A&M University System, College Station, TX, 1993.

[6] Stephen G. Ritchie and Baher Abdulhai. Development testing and evaluation of advanced techniques for freeway incident detection. Institute of transportation studies, research reports, working papers, proceedings, Institute of Transportation Studies, UC Berkeley, January 1997.

[7] Tomáš Šingliar and Miloš Hauskrecht. Learning to detect adverse traffic events from noisily labeled data. In *Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2007, pages 236–247, Berlin, Heidelberg, 2007. Springer-Verlag.

[8] H. J. Payne, E. D. Helfenbein, and H. C. Knobel. Development and testing of incident detection algorithms, volume 1: Summary of results. Technical Report FHWA-RD-76-19, Federal Highway Administration, U.S. Department of Transportation, April 1976.

[9] H. J. Payne, E. D. Helfenbein, and H. C. Knobel. Development and testing of incident detection algorithms, volume 2: Research methodology and detailed results. Technical Report FHWA-RD-76-20, Federal Highway Administration, U.S. Department of Transportation, April 1976.

[10] H. J. Payne and S. C. Tignor. Freeway incident-detection algorithms based on decision trees with states. *Transportation Research Record: Journal of the Transportation Research Board*, 682:30–37, 1978.

[11] P. H. Masters, J. K. Lam, and Kam Wong. Incident detection algorithms for COMPASS - an advanced traffic management system. In *Vehicle Navigation and Information Systems Conference, 1991*, volume 2, pages 295–310, 1991.

[12] J. F. Collins, C. M. Hopkins, and J. A. Martin. Automatic incident detection: TRRL algorithms HIOCC and PATREG. Technical Report IRRD 246992, Transport and Road Research Laboratory, Berkshire, England, 1979.

[13] C. L. Dudek, C. J. Messer, and N. B. Nuckles. Incident detection on urban freeway. *Transportation Research Record: Journal of the Transportation Research Board*, 495:12–24, 1974.

[14] M. Levin and G. M. Krause. Incident detection: A bayesian approach. *Transportation Research Record: Journal of the Transportation Research Board*, 682:52–58, 1978.

[15] George Edward Pelham Box and Gwilym Jenkins. *Time Series Analysis, Forecasting and Control.* Holden-Day, Incorporated, 1990. ISBN 0816211043.

[16] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer, 2nd edition, March 2002. ISBN 0387953515.

[17] M. S. Ahmed and A. R. Cook. Analysis of freeway traffic time-series data using Box-Jenkins techniques. *Transportation Research Record: Journal of the Transportation Research Board*, 722:1–9, 1977.

[18] M. S. Ahmed and A. R. Cook. Time series models for freeway incident detection. *Transportation Engineering Journal*, 106:731–745, 1980.

[19] M. S. Ahmed and A. R. Cook. Application of time-series analysis techniques to freeway incident detection. *Transportation Research Record: Journal of the Transportation Research Board*, 722: 1–9, 1977.

[20] Allen R. Cook and Donald E. Cleveland. Detection of freeway capacity-reducing incidents by traffic-stream measurements. *Transportation Research Record: Journal of the Transportation Research Board*, 495:1–11, 1974.

[21] Y. J. Stephanedes and A. P. Chassiakos. Application of filtering techniques for incident detection. *Journal of Transportation Engineering*, 119(1):13–26, 1993.

[22] Yorgos J. Stephanedes and Athanassios P. Chassiakos. Freeway incident detection through filtering. *Transportation Research Part C: Emerging Technologies*, 1(3):219–233, 1993.

[23] Y. J. Stephanedes and A. P. Chassiakos. Smoothing algorithms for incident detection. *Transportation Research Record: Journal of the Transportation Research Board*, 1394:8–16, 1993.

[24] A. Samant and H. Adeli. Feature extraction for traffic incident detection using wavelet transform and linear discriminant analysis. *Computer-Aided Civil and Infrastructure Engineering*, 15(4):241–250, 2000.

[25] Ruey L. Cheu, S. G. Ritchie, W. W. Recker, and B. Bavarian. Investigation of a neural network for freeway incident detection. Technical Report UCI-ITS-WP-91-6, University of California, Irvine, 1991.

[26] Stephen G. Ritchie and Ruey L. Cheu. Simulation of freeway incident detection using artificial neural networks. *Transportation Research Part C: Emerging Technologies*, 1(3):203 – 217, 1993.

[27] Ruey L. Cheu and Stephen G. Ritchie. Automated detection of lane-blocking freeway incidents using artificial neural networks. *Transportation Research Part C: Emerging Technologies*, 3(6): 371 – 388, 1995.

[28] Y. J. Stephanedes and X. Liu. Artificial neural networks for freeway incident detection. *Transportation Research Record: Journal of the Transportation Research Board*, 1494:91–97, 1995.

[29] Hussein Dia and Geoff Rose. Development and evaluation of neural network freeway incident detection models using field data. *Transportation Research Part C: Emerging Technologies*, 5 (5):313 – 331, 1997.

[30] Baher Abdulhai and Stephen G. Ritchie. Enhancing the universality and transferability of freeway incident detection using a bayesian-based neural network. *Transportation Research Part C: Emerging Technologies*, 7(5):261 – 280, 1999.

[31] Baher Abdulhai and Stephen G. Ritchie. Preprocessor feature extractor and post processor probabilistic output interpreter for improved freeway incident detection. *Transportation Research Record: Journal of the Transportation Research Board*, 1678:277–286, 1999.

[32] Xin Jin, Ruey Long Cheu, and Dipti Srinivasan. Development and adaptation of constructive probabilistic neural network in freeway incident detection. *Transportation Research Part C: Emerging Technologies*, 10(2):121 – 147, 2002.

[33] Prasenjit Roy and Baher Abdulhai. Gaid: Genetic adaptive incident detection for freeways. *Transportation Research Record: Journal of the Transportation Research Board*, 1856:96–105, 2003.

[34] H. Adeli and A. Karim. Fuzzy-wavelet rbfnn model for freeway incident detection. *Journal of Transportation Engineering*, 126(6):464–471, 2000.

[35] A. Karim and H. Adeli. Comparison of fuzzy-wavelet radial basis function neural network freeway incident detection model with california algorithm. *Journal of Transportation Engineering*, 128(1):21–30, 2002.

[36] A. Karim and H. Adeli. Incident detection algorithm using wavelet energy representation of traffic patterns. *Journal of Transportation Engineering*, 128(3):232–242, 2002.

[37] A. Karim and H. Adeli. Fast automatic incident detection on urban and rural freeways using wavelet energy algorithm. *Journal of Transportation Engineering*, 129(1):57–68, 2003.

[38] H. Adeli and A. Samant. An adaptive conjugate gradient neural networkwavelet model for traffic incident detection. *Computer-Aided Civil and Infrastructure Engineering*, 15(4):251–260, 2000.

[39] A. Samant and H. Adeli. Enhancing neural network traffic incident-detection algorithms using wavelets. *Computer-Aided Civil and Infrastructure Engineering*, 16(4):239–245, 2001.

[40] Xueqing Cheng, Wenfang Lin, Enxiang Liu, and Dan Gu. Highway traffic incident detection based on BPNN. *Procedia Engineering*, 7(0):482 – 489, 2010.

[41] D. Srinivasan, V. Sharma, and K.A. Toh. Reduced multivariate polynomial-based neural network for automated traffic incident detection. *Neural Networks*, 21(23):484 – 492, 2008.

[42] Dipti Srinivasan, Xin Jin, and Ruey Long Cheu. Evaluation of adaptive neural network models for freeway incident detection. *Intelligent Transportation Systems, IEEE Transactions on*, 5 (1):1–11, 2004.

[43] Dipti Srinivasan, Xin Jin, and Ruey Long Cheu. Adaptive neural network models for automatic incident detection on freeways. *Neurocomputing*, 64:473–496, March 2005.

[44] Edmond Chin-Ping Chang. Fuzzy systems based automatic freeway incident detection. In *1994 IEEE International Conference on Systems, Man, and Cybernetics, 1994. Humans, Information and Technology*, volume 2, pages 1727–1733 vol.2, 1994.

[45] E. C.-P. Chang and S. H. Wang. Improved freeway incident detection using fuzzy set theory. *Transportation Research Record: Journal of the Transportation Research Board*, 1453:75–82, 1994.

[46] C.-K. Lin and G.-L. Chang. Development of a fuzzy-expert system for incident detection and classification. *Mathematical and Computer Modelling*, 27(911):9 – 25, 1998.

[47] Kong Yaguang and Xue Anke. Urban traffic incident detection based on fuzzy logic. In *32nd Annual Conference on IEEE Industrial Electronics, IECON 2006*, pages 772–775, 2006.

[48] Yaser E. Hawas. A fuzzy-based system for incident detection in urban street networks. *Transportation Research Part C: Emerging Technologies*, 15(2):69 – 95, 2007.

[49] Xie Binglei, Hu Zheng, and Ma Hongwei. Fuzzy-logic-based traffic incident detection algorithm for freeway. In *Machine Learning and Cybernetics, 2008 International Conference on*, volume 3, pages 1254–1259, 2008.

[50] Shen Xiaorong, Zhang Hai, Fan Yaozu, and Xu Chao. Fuzzy learning classifier system and its application research in automatic traffic incident detection. In *2nd IEEE Conference on Industrial Electronics and Applications, 2007. ICIEA 2007*, pages 769–772, 2007.

[51] C. Hsiao, C. Lin, and M. Cassidy. Application of fuzzy logic and neural networks to automatically detect freeway traffic incidents. *Journal of Transportation Engineering*, 120(5):753–772, 1994.

[52] Seung-Heon Lee, Jin-Woo Choi, Nam-Kwan Hong, M. Viswanathan, and Young-Kyu Yang. Development of incident detection model using neuro-fuzzy algorithm. In *Computer and Information Science, 2005. Fourth Annual ACIS International Conference on*, pages 364–368, 2005.

[53] M. Viswanathan, S. H. Lee, and Y. K. Yang. Neuro-fuzzy learning for automated incident detection. In *Proceedings of the 19th international conference on Advances in Applied Artificial Intelligence: industrial, Engineering and Other Applications of Applied Intelligent Systems*, IEA/AIE'06, pages 889–897. Springer-Verlag, 2006.

[54] Sherif S. Ishak and H. M. Al-Deek. Freeway incident detection using fuzzy ART. In *Fifth International Conference on Applications of Advanced Technologies in Transportation Engineering*, pages 59–66, 1998.

[55] Sherif S. Ishak and H. M. Al-Deek. Fuzzy art neural network model for automated detection of freeway incidents. *Transportation Research Record: Journal of the Transportation Research Board*, 1634:56–63, 1998.

[56] D. Srinivasan, S. Sanyal, and Woei Wan Tan. Hybrid neuro-fuzzy technique for automated traffic incident detection. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pages 713–719, 2006.

[57] Fang Yuan and Ruey Long Cheu. Incident detection using support vector machines. *Transportation Research Part C: Emerging Technologies*, 11(34):309 – 328, 2003.

[58] Shuyan Chen, Wei Wang, and Henk van Zuylen. Construct support vector machine ensemble to detect traffic incident. *Expert Systems with Applications*, 36(8):10976 – 10986, 2009.

[59] Jianli Xiao and Yuncai Liu. Traffic incident detection by multiple kernel support vector machine ensemble. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 1669–1673, 2012.

[60] Jianli Xiao, Xiang Gao, Qing-Jie Kong, and Yuncai Liu. More robust and better: a multiple kernel support vector machine ensemble approach for traffic incident detection. *Journal of Advanced Transportation*, pages n/a–n/a, 2013.

[61] Dehuai Zeng, Jianmin Xu, and Gang Xu. Data fusion for traffic incident detector using d-s evidence theory with probabilistic svms. *Journal of Computers*, 3(10), 2008.

[62] E. Parkany and D. Bernstein. Design of incident detection algorithms using vehicle-to-roadside communication sensors. *Transportation Research Record: Journal of the Transportation Research Board*, 1494:67–74, 1995.

[63] Vaneet Sethi, Nikhil Bhandari, Frank S. Koppelman, and Joseph L. Schofer. Arterial incident detection using fixed detector and probe vehicle data. *Transportation Research Part C: Emerging Technologies*, 3(2):99 – 112, 1995.

[64] M.William Sermons and Frank S. Koppelman. Use of vehicle positioning data for arterial incident detection. *Transportation Research Part C: Emerging Technologies*, 4(2):87 – 96, 1996.

[65] K. F. Petty, A. Skabardonis, and P. P. Varaiya. Incident detection with probe vehicles: performance, infrastructure requirements and feasibility. In *Proceedings Volume from the 8th IFAC/IFIP/IFORS Symposium on Transportation Systems*, pages 125–130, Chania, Greece, June 1997.

[66] K. C. Mouskos, E. Niver, S. Lee, T. Batz, and P. Dwyer. Transportation operations coordinating committee system for managing incidents and traffic evaluation of the incident detection system. *Transportation Research Record: Journal of the Transportation Research Board*, 1679: 50–57, 1999.

[67] E. Niver, K.C. Mouskos, T. Batz, and P. Dwyer. Evaluation of the TRANSCOM's system for managing incidents and traffic (TRANSMIT). *Intelligent Transportation Systems, IEEE Transactions on*, 1(1):15–31, 2000.

[68] B. Hellinga and G. Knapp. Automatic vehicle identification technology-based freeway incident detection. *Transportation Research Record: Journal of the Transportation Research Board*, 1727:142–153, 2000.