th

WCTR

IDENTIFYING MOTORWAY INCIDENTS BY NOVELTY DETECTION

HAIBO CHEN
Institute for Transport Studies
University of Leeds
Leeds LS2 9JT, UK

ROGER D. BOYLE
School of Computer Studies
University of Leeds
Leeds LS2 9JT, UK

HOWARD R. KIRBY
Transport Research Institute
Napier University
Redwood House, 66 Spylaw Road, Edinburgh, EH10 5BR, UK

FRANK O. MONTGOMERY
Institute for Transport Studies
University of Leeds
Leeds LS2 9JT, UK

Abstract

The class of real interest (e.g. traffic incidents) is under-represented in our database. This is because the abnormalities are rare and difficult to collect in safety-critical applications. Conventional incident detection algorithms focus on identification and characterisation of a wide range of different incidents from historic data. In this paper, alternative approaches which estimate the probability density of incident-free data are proposed. The abnormality of an input vector is therefore identified by testing for novelty against the description of normality. Experimental results on both simulated and field data show that the techniques are capable of detecting incidents and can thus be used in dynamic traffic monitoring systems.

INTRODUCTION

Severe traffic congestion can cause loss of accessibility, environmental pollution and energy wastage. There are two common types of congestion. The first is due to increases in demand above the current capacity of the road system; the second is due to reductions in capacity below the current demand levels. The first is typified by recurrent congestion, and is due to variations in the demand for use of the road. Such variability, from one season of the year or day of week to another can easily be predicted. To reduce the possibility of such congestion, traffic management and control techniques such as variable message signs (VMS) and ramp-metering have been introduced. In a more recent development, the UK Highways Agency has implemented mandatory variable speed limits on the M25 orbital road around London. These strategies help to smooth out the demand to a level below network capacity. The second type of congestion, due to capacity unexpectedly falling, typically arises due to an accident or traffic incident. The occurrence of an incident is essentially an unpredictable event, and so techniques are needed to detect them.

As traffic flow increases, incidents, though inherently rare, are becoming more common and more disruptive. Therefore, the development of effective and reliable incident detection techniques is becoming increasingly necessary for the efficient management of traffic systems. Whilst traffic monitoring and surveillance techniques have improved dramatically over recent years, with rich streams of data becoming routinely available to traffic operators, the ability to interpret these data, and to yield robust and reliable indicators that an incident has occurred, has become a challenging problem. Ideally one needs an incident detection system that always alerts an operator to a serious incident; never provides a false alert; and gives appropriate warnings in intermediate situations.

In the UK, the incident detection methodology most widely used on motorways derives from the "HIOCC" algorithm developed by the then Transport and Road Research Laboratory (Collins *et al*, 1979). As its name suggests, it detects periods when the "occupancy" measure (proportion of time that an inductive loop is occupied by vehicles) is particularly high. The occupancy measure is calculated in each one-tenth second interval, but uses data from a single site. A number of other incident detection techniques have been developed for motorway traffic management systems. Examples include the California Algorithm (Payne *et al*, 1976); Probabilistic Approaches (Levin and Krause, 1978); Time Series Analysis (Ahmed and Cook, 1982); Fuzzy Logic (Chang and Wang, 1994); Catastrophe Theory (Persaud and Hall, 1989; Forbes, 1992), upon which the McMaster Algorithm is based; and Neural Networks (Cheu and Ritchie, 1995).

By contrast with the HIOCC approach, many of these other techniques integrate data from more than one site. In principle, neural networks are a particularly appropriate tool for this, since they can cope with simultaneous streams of data from several sources. Incident detection is essentially a matter of pattern recognition, which is what neural networks are good at doing. They have shown outstanding performance in dealing with large, noisy and incomplete input patterns.

However, the most common type of neural network, described as "supervised", needs to be trained to recognise the different patterns. Supervised networks require a wide range of different examples to be included in the training data sets. In the domain of traffic analysis, it is nowadays easy to acquire vast volumes of data on flows, speeds and occupancies from different sites, and supervised networks have been successfully used for such tasks as short term traffic forecasting (Clark *et al*, 1993; Dougherty and Cobbett, 1994; Smith and Demetsky, 1994). But for the incident detection task, there is a problem: incidents are rare events, and obtaining a large number of examples of incidents in the training set would be a time consuming and difficult task, especially as their location and time of occurrence are often not recorded precisely. Whilst, in some other application areas (for example, equipment failures), it is possible to inflate the number of unusual events artificially, that is not exactly appropriate for traffic accidents and incidents! Cheu and Ritchie (1995) circumvented the problem by training their supervised neural network (which was based on a multi-layer feed-forward architecture) on simulated data, generated by a micro-simulation traffic model. With this they artificially generated enough incidents to train the supervised neural network. However, it is highly unlikely or at least difficult to calibrate a simulation model exactly to real traffic. In other words, the techniques which work well on simulated data may not do so on real data. Moreover, since the fractions of examples from different classes in the training data set differ from the real situation, failure to take correct account of their differing prior probabilities can lead to significantly sub-optimal results when applied to the real world.

Instead of distinguishing between normal traffic and incident-affected traffic, an alternative approach is to build up a model that describes the distribution of traffic under normal circumstances; and thus to identify an incident as a novel departure from this distribution. This approach is described as *Novelty Detection*. Such an approach is able to discriminate a novel input from normal ones by measuring its distance to the centroid of the distribution of the normal data. Such data outliers clearly indicate that something unusual is taking place, and so should include all those incidents that affect the normal flow of traffic. Clearly, the outliers might also identify non-incident events (such as unusually heavy traffic generated by a Trade Show, or a police chase); but to the traffic operator, it may be just as important to know about these as about traffic incidents. The focus on novelty detection rather than incident detection makes it much easier to develop appropriate models, since they can be based on readily acquired normal data. Instead of supervised neural networks, unsupervised ones can be used, as can a number of other related techniques, which we now review.

POSSIBLE APPROACHES FOR NOVELTY DETECTION

For novelty detection to be possible, two steps are needed. Firstly, one needs to be able to characterise the distribution of the normal data, by some form of probability density function. Secondly, a measure of novelty has to be defined. The two issues are discussed in more detail for three possible novelty detection approaches as follows.

However, the specification of the probability density function is not an easy matter. That is because the mean (of any variable, such as speed, flow or occupancy) is changing across the data-set, as is the distribution about that mean. So one regards the data as having been generated by a number of generators, each characterised by its own statistical distribution. The usual practice is to divide the normal data into clusters, the number of these depending on the complexity of the data. In addition, the discriminant surfaces or decision boundaries of these clusters must be formed so that the novelty of an input vector can be measured.

K-means algorithm

Parametric estimation of the probability density function requires a priori statistical information such as the number of generators and the underlying functions within the data set. For most real-world applications, however, the information is unknown. Non-parametric techniques, such as Parzen windows (Parzen, 1962), solve this problem by obtaining an estimate of density at every data point in the training set. In this way the form of the density is determined entirely by the data. A kernel function is defined to estimate the probability density of data points within each window; and discontinuities in the density estimates can be smoothed by the use of a suitable form of kernel function (Bishop, 1995). However, these techniques are computationally expensive for large data sets and have the disadvantage of modelling any noise in the training set. Therefore a more parsimonious representation of the training data is desirable.

Semi-parametric estimation achieves this more parsimonious representation by applying kernel estimation methods, as used in the non-parametric techniques, to a smaller number of kernel functions. The number is less than the number of data points but still large compared to the probable number of generators within the data. In many practical applications, this approach has been shown to be an effective method for density estimation (Traven, 1991 and Ripley, 1992). Semi-parametric estimation may be regarded as a clustering or partitioning of the data set in terms of cluster means and covariance matrices. The number of kernel functions used by the K-means algorithm must be decided in advance. The calculation of the centres and membership of the clusters associated with each the kernel functions can be implemented as follows.

1) Define the number K (say K = 4) of kernel functions/clusters.
2) Assign the data points at random to one of the K sets.
3) Compute the mean of the data points in each set.
4) Re-assign each data point to a new set according to which is the nearest mean.
5) Re-compute the means of the sets.
6) Repeat steps 4 and 5 until there is no further change in the grouping of the data points.

The major problem with any form of novelty detection is the choice of an appropriate "novelty threshold" (Roberts and Tarassenko, 1994). For traffic incident detection, too low a threshold may cause too many false alarms whereas a high threshold may generate a low detection rate. The determination of a suitable novelty threshold may thus be empirical and application-oriented. A simple way to define the novelty threshold is to set it at the 'normality' value of the most abnormal data point in the training set. The novelty of an input data vector, $\mathbf{x}_j$, can be measured by the shortest normalised distance of the input vector to a cluster centre, $\mathbf{c}_i$, as follows:

$$d_j = \min_i \frac{1}{\bar{d}_i} \sqrt{(\mathbf{X}_j - \mathbf{C}_i)^2} \qquad (1)$$

where: $\bar{d}_i$ is the average distance between all the data vector belonging to a cluster and the cluster centre.

Principal component analysis

Principal Component Analysis (PCA) is a well-known statistical technique widely used in data analysis, for purposes such as feature extraction, data compression and multivariate data projection (Oja, 1992). It is a linear orthogonal transform from an L-dimensional input space to an M-dimensional feature space, $M \le L$, such that the co-ordinates of the data in the new M-dimensional space are un-correlated and maximal variance of the original data is preserved by only a small number of co-ordinates. This may imply that the intrinsic dimension of the original data is much less than that used.

In our first data set generated by a traffic simulation model, if the inputs are the flow, speed and occupancy from upstream and downstream detectors, then L is 6. The kth L-dimensional input vector is denoted as

$$\mathbf{x}_k = \left(Q_k^u, U_k^u, O_k^u, Q_k^d, U_k^d, O_k^d\right)^T \qquad (2)$$

where $k = 1, 2, \ldots, n$;
n is the number of observations;
Q, U and O denote flow, speed and occupancy respectively;
u and d stand for upstream and downstream respectively.

The $\mathbf{x}_k$ are then normalised to have zero mean and unit standard deviation.

Let C be the covariance matrix of the input data, with eigenvalues, $\lambda_1, \lambda_2, \cdots, \lambda_L$, in decreasing order of magnitude. The M principal components of the input data can be obtained by a linear transform

$$b_k = x_k e(k) \tag{3}$$

where b_k is an n x M matrix
$e(k)$ is an L x M matrix whose columns are M eigenvectors corresponding to the M largest eigenvalues of the covariance matrix C.

Eqn (3) implements the transform from the original data space X to principal component space B. The dimension of b is normally much less than that of x without loss of information. This would mask out the small-scale noise effects. Another advantage of the technique is that PCA could be implemented directly in a neural network architecture that possesses high parallel computation rates (Diamantaras and Kung, 1996).

For the traffic incident detection by novelty detection techniques, the eigenvalues and the eigenvectors are calculated on the incident-free traffic data to form the representation of the normal data set.

Normally the first few principal components would represent the absolute majority of information in the original data. The sum of absolute values of m principal components is shown below:

$$y_k = \sum_{i=1}^{m} |b_{ki}| \tag{4}$$

where y_k is the kth transformed value corresponding to $\mathbf{x}_k$.

The novelty threshold is set at the maximum value of the $\mathbf{y}_k$ in the training set.

Auto-associative neural network

An auto-associative neural network is simply a Multi-Layer Perceptron (MLP) that is trained to map input vectors onto themselves by minimisation of a sum-of-squares error (Bishop, 1995). Since no independent target data is provided, the auto-associative network performs unsupervised training.

In a multi-layer perceptron the processing units or neurons are distributed into several layers: the input layer, the output layer and some hidden layers in between. The connection weights between the two adjacent layers are adjustable. With such an architecture, a complicated modelling operation can be broken into a number of much simpler sub-operations and distributed in the layers of a network. The sub-operations in the same layer normally take the same inputs and therefore can be implemented in parallel.

The input layer usually operates as an input data receiver which distributes the external signals to the first hidden layer. The signals are then propagated forward to the following hidden layers if these exist, and finally arrive in the output layer which provides the network output signal. This normally involves two operations. The first is to produce an activation for the neuron by combining the outputs of the nodes of the preceding layer, by using the following formulation:

$$z_j^k(t) = \sum_{i=1}^{N_{k-1}} w_{ij}^k \cdot x_i^{k-1}(t) + b_j^k(t) \tag{5}$$

where t denotes the signal being processed;

$z_j^k(t)$ is the activation of the jth node at the kth layer;

w_{ij}^k is the connection weight between the ith neuron of the (k-1)th layer and the jth neuron of the kth layer;

$x_i^{k-1}(t)$ is the output of the ith neuron of the (k-1) th layer;

$b_j^k(t)$ is the bias or the threshold of the neuron;

N_{k-1} is the number of neurons in the (k-1)th layer.

The second operation, termed the activation operation, produces the output $x_j^k(t)$ of a neuron given its activation, $z_j^k(t)$. It is common practice for the activation function to be linear, sigmoid or hyperbolic tangent.

Training a neural network involves identifying those connection weights which minimise the difference between the network output and the desired output (e.g. targets). The most common method of adjusting the connection weights is to minimise the mean square error in output activation. It is easily computed, has proven itself in practice, and perhaps most importantly, its partial derivative with respect to individual weights can be computed explicitly. The mean square error is found by averaging, over the training data set, the square of the difference between the individual outputs of the network and their targets. The error E is then back-propagated through the network by a gradient descent method. The partial derivative of the error with respect to each weight determines whether the relevant weight will be adjusted upwards or downwards in order to reduce the error. The updating of a weight at each iteration t is carried out by adding the increment to the weight. The increment is given by:

$$\Delta w_{ij}^k(\tau) = \sum_t \eta \delta_j^k(t) x_i^{k-1}(t) + \mu \Delta w_{ij}^k(\tau - 1) \tag{6}$$

where η is the learning rates which determines the influence of the error signal on the value of the parameter changes;

μ is a momentum constant which determines the influence of the past parameter changes on the current direction;

$\delta_j^k(t)$ is the error signal of the jth neuron of the kth layer which is back-propagated in the network.

For the neurons in the hidden layer, the error signal can be expressed as:

$$\delta_j^k(t) = f'(x_j^k(t)) \sum_{i=1}^{N_{k+1}} \delta_i^{k+1}(t) w_{ji}^{k+1}(\tau - 1) \tag{7}$$

The targets used to train the auto-associative network are simply the input vectors themselves, so that the network is attempting to map each input vector onto itself. Such a network is then thought of as being equivalent to linear principal component analysis (Bishop, 1995).

The relative RMS error is calculated at the output of auto-associative network when an input vector is present. The maximum relative RMS in the training set is chosen as the novelty threshold for the model.

THE DATA

Initial experiments were carried out with simulated data; before progressing to tests with real data.

Simulated data

For trials with simulated data, use was made of a test-bench already developed for another project at ITS University of Leeds (Shepherd and Chen, 1996). The test-bench represented the Kent motorway corridor. The area covered includes junctions 1 - 5 of the M2, junctions 4 - 8 of the M20, (both lengths about 20 km), and the main interconnecting A class roads. The test-bed was developed by using the micro-simulation model AIMSUN2. Simulated detectors were implemented at 500m intervals along the M2 and M20. These detectors provided one-minute summaries of flow, speed and occupancy for the simulated flows across the carriageway. The simulated network was calibrated to flows provided by TRL from an MCONTRM model of the area. Traffic consists of: small car 37%, medium car 28%, large car 25% and lorry 10%. The network was simulated for 3 hours (7:30 - 10:30) each time. To generate different levels of traffic, the entry flows to the network were factored by 0.9, 1.1, 1.2, 1.3 and 1.4.

To investigate novelty-detection methodology, incidents were placed on the Eastbound section of the M2 between the first two detectors, just after junction 5. Eighteen one-lane and all-lane blocking incidents were generated having a random start time and random duration of between 8 to 20 minutes under different traffic demands. The average duration and standard deviation of an incident were taken from Roberts *et al* (1994). Four extra incidents were placed at the upstream/downstream sections to evaluate the effects of upstream/downstream disturbances. In addition, three heavy congestion scenarios were generated by slowing the traffic at the downstream section and factoring the entry flows to the network by 1.6. Five incident-free simulation runs under various traffic demands were implemented. This set of data, comprising 30 simulation runs in all, was then used to estimate the probability density function of the normal traffic data.

Real data from the M25

Real-time traffic data is inevitably corrupted by high levels of noise and unexpected disturbances such as detector faults and transmission distortion, and this influences the design of successful incident detection techniques. Simultaneously, the data load may be very high. This project uses data from the M25 motorway – an orbital highway around London. The test site was chosen to be between junctions 10 and 15, where there has been extensive instrumentation. The traffic on each lane of the road is monitored by loop detectors placed approximately 500 metres apart. The data obtained consists of minute-by-minute estimates of flow (vehicles/hour), speed (km/hour) and occupancy (point density, % of time occupied by vehicles). The section under study generates 1820 measurements per minute for a 28.8 kilometre section of highway.

The training data was collected between 1 and 14 March 1997 over the time periods between 12:00 and 16:00 so that it mainly contains normal traffic. The lane-based traffic variables were averaged across the carriageway. The parameters of the proposed models were estimated over the normal data rather than incidents.

In order to identify the location of an incident it is common to consider two adjacent detectors that would monitor the traffic behaviour between them. By applying the same arrangement to any pair of the adjacent detectors one would monitor the traffic at any site on the road.

EXPERIMENTS AND DISCUSSIONS

Data re-scaling

Traffic measurements (*i.e.* flow, speed and occupancy) used in the work have typical values which (depending on the units in which each of them is expressed) differ by several orders of magnitude. Furthermore, the typical sizes of the measurements may not reflect their relative importance in determining the required outputs. Therefore for the techniques used in this work the traffic measurements need to be re-scaled to have similar values. *Component-wise normalisation* and *whitening* are commonly used (Bishop, 1995). The whitening technique removes any correlation between the measurements but component-wise normalisation does not. Both techniques yield re-scaled measurements that have zero mean and unit variance. Recently, a more efficient approach based on entropy maximisation has been proposed by one of the authors (Boyle, 1997). Instead of normalising the measurements to have zero mean and unit variance, the new method tries to find the scalar for a new measurement so that the effect of reducing the descriptive power of some data components as possessed by the former two techniques would be removed. The experimental comparison between the techniques is being studied but, in this paper, only component-wise normalisation is used.

Results on simulated data

The performances of the three models are measured by the three criteria: Detection Rate (DR), False Alarm Rate (FAR) and Time To Detection (TTD). The definitions of these criteria are well described in earlier work (Payne and Tignor, 1978; Levin and Krause, 1979; Hall *et al.*, 1994). Figure 1 shows the outputs of the models versus time on a simulated incident run. (Note that the y-axes are re-scaled so that the outputs of each model have a maximum of 1). The simulation run implements an incident which starts at 9:00 and lasts for 20 minutes. All the models respond to the incident at a similar time. However, the auto-associative neural network has higher fluctuations during the incident period and an earlier end-point than other models.

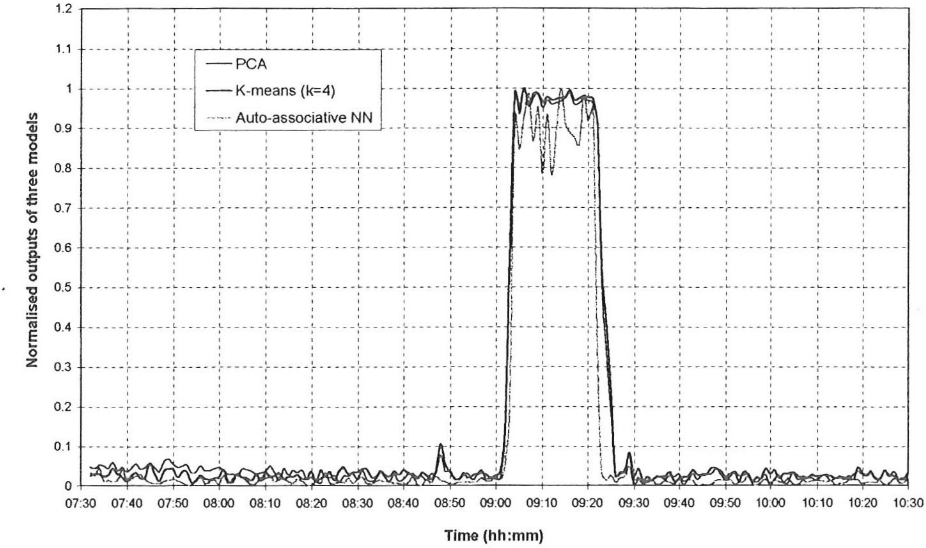

Figure 1 - The normalised outputs of the three models on simulated incident data.

Table 1 summarised the detection performance of the models on all test data sets. All simulated incidents were detected by the three models. The K-means algorithm has faster detection but a higher FAR than others.

Table 1 - Results on simulated M2 data

Techniques	DR(%)	FAR(%)	Average TTD(sec.)
PCA	100	1.10*	106.67
K-means (k=4)	100	2.42*	70.00
Auto-associative NN	100	1.28*	106.67

Note1: DR: Detection Rate; FAR: False Alarm Rate; TTD: Time To Detection.

Note2: * indicates that all false alarms were caused by queues propagated by incidents outside the test section. These could be eliminated if the algorithm was implemented on each section.

Results on real M25 data

The models were applied to real M25 traffic data. The test results of the models are shown in Figures 2, 3 and 4 respectively. The inferred incident occurred at 16:36 on 7th August 1996 and lasted for 36 minutes. Twenty-four lane-by-lane flow, speed and occupancy estimates from upstream and downstream detectors across 4 lanes were used as inputs to the system.

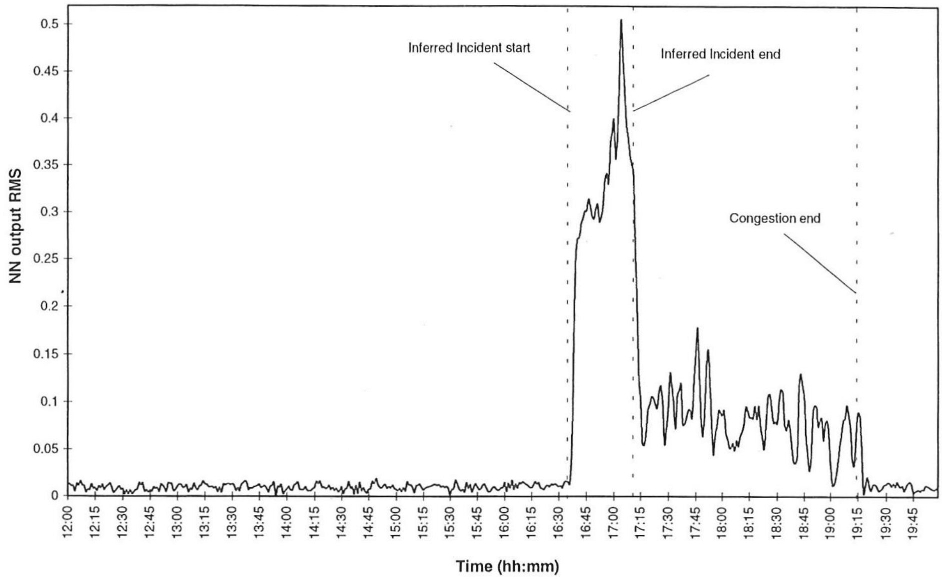

Figure 2 - The relative RMS of the outputs of the auto-associative MLP versus time

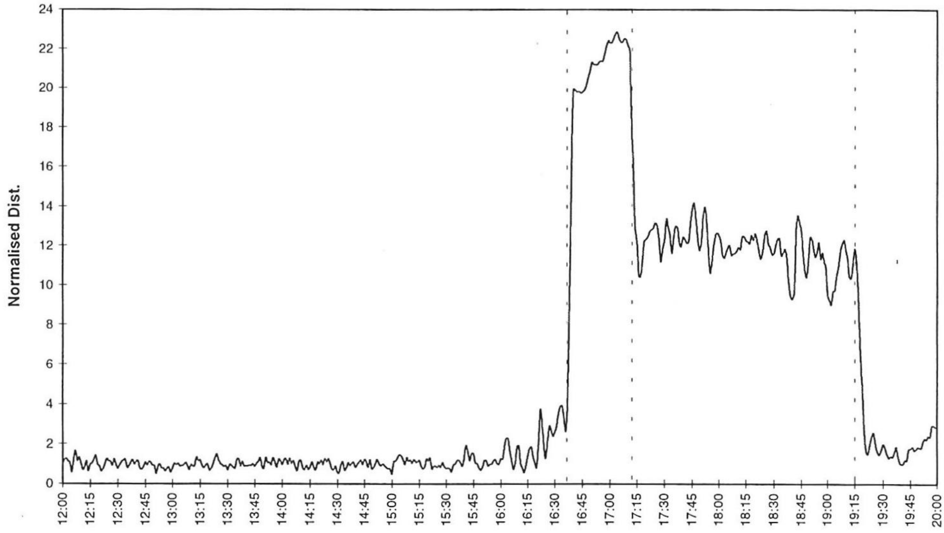

Figure 3 - The normalised distance of inputs to the nearest centre by K-means algorithm

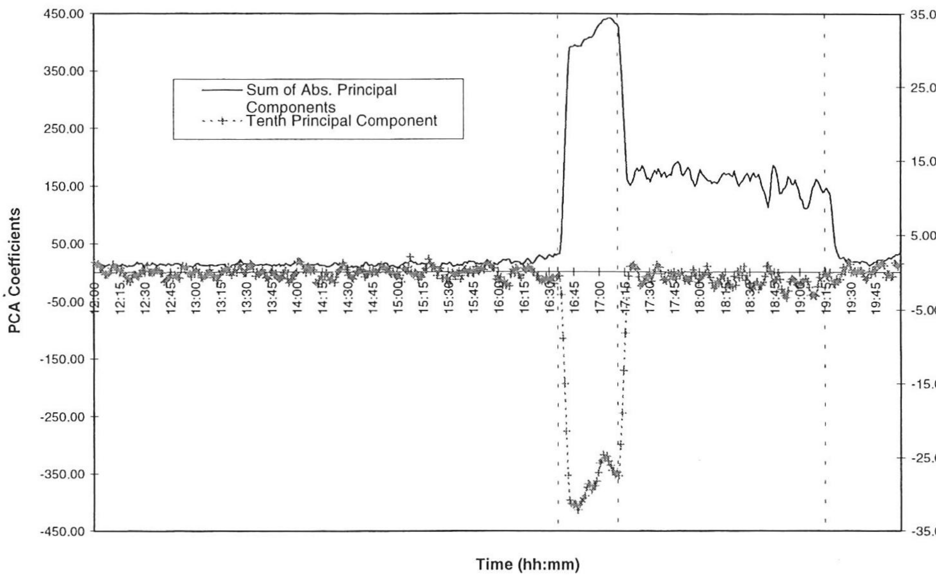

Figure 4 - The Principal Components (PCs)

The results show that all the models are capable of detecting an abnormal event. Three different traffic behaviours can be recognised by the magnitudes of the output values of the models. The lowest values would correspond to the free-flow traffic whereas the highest ones represent the incident data. A period of moderate but fluctuating values following the incident was thought of as the recovery from the incident. Figure 4 also shows that the tenth principal component have strong response to the incident period.

CONCLUSIONS

The use of novelty detection for traffic incident identification has been investigated. Instead of modelling the difference between normal traffic and incidents, The K-means algorithm, PCA and auto-associative neural network have been trained only on normal traffic data.

The performances of the three models have been compared in terms of detection rate, false alarm rate and time to detection on simulated data from M2 in the Kent corridor. All simulated incidents were detected by the three models. It was shown that the K-means algorithm has faster detection but a higher false alarm rate than the other two. It should be noted that all false alarms generated by the models were caused by queues propagated by incidents outside the test section. These could be eliminated if the algorithm was implemented on each section.

An initial study using real traffic data from M25 has shown that all the models are capable of discriminating different traffic patterns. Work continues on further developments and validation of the models on more real traffic incidents.

ACKNOWLEDGEMENTS

The authors would like to thank EPSRC for their financial support, which was under the *Neural Computing – The Key Questions* directed programme, and the Highways Agency for providing the M25 traffic data.

REFERENCES

Ahmed, S.R. and Cook, A.R. (1982) Application of time-series analysis techniques to freeway incident detection. **Transportation Research Record 841**, 19-21.

Bishop, C.M. (1995) **Neural Networks for Pattern Recognition**. Clarendon Press, Oxford.

Boyle, R.D. (1998) Scaling additional contributions to {Principal Components Analysis}. **Pattern Recognition,** Vol. 31, No. 12, 2047-2053.

Chang, E.C. and Wang, S. (1994) Improved Freeway incident Detection using Fuzzy Set Theory. **Transportation Research Record 1453**, 75-82.

Cheu, R.L. and Ritchie, S.G. (1995) Automated Detection of Lane-Blocking Freeway Incidents using Artificial Neural Networks. **Transportation Research 3C** (6), 371-88.

Clark, S.D., Dougherty, M.S. and Kirby, H.R. (1993) The use of neural networks and time series models for short term traffic forecasting: a comparative study. **Proceeding of the PTRC summer meeting**, Vol. 363, 151-162.

Collins, J.F., Hopkins C.M. and Martin J.A. (1979) Automatic incident detection - TRRL algorithms HIOCC and PATREG. **TRRL Supplementary Report,** No. 526, Crowthorne, Berkshire, UK.

Diamantaras, K.I. and Kung, S.Y. (1996) **Principal Component Neural Networks - Theory and Applications**. John Wiley & Sons, New York.

Dougherty, M. and Cobbett, M. (1994) Short term inter-urban traffic forecasts using neural networks. **Proceedings of the 2nd DRIVE II Workshop on Short Term Traffic Forecasting**, Delft, 65-80.

Hall, F.L., Shi, Y. and Atala, G. (1994) On-line Testing of the McMaster Incident Detection Algorithm Under Recurrent Congestion. **Transportation Research Record 1394**, 1-7.

Levin, M. and Krause, G.M. (1978) Incident detection: A bayesian approach. **Transportation Research Record 682**, 52-58.

Oja, E. (1992) Principal components, minor components, and linear neural networks. **Neural Networks**, Vol. 5, 927-935.

Parzen, E. (1962) On Estimation of a Probability Density Function and Mode. **Annals of Mathematical Statistics 33**, 1065-1076.

Payne, H.J. and Tignor, S.C. (1978) Freeway incident-detection algorithm based on decision trees with states. **Transportation Research Record 682**, 30-37.

Persaud, B. and Hall, F.L. (1989) Catastrophe theory and patterns in 30-second freeway traffic data - Implications for incident detection. **Transportation Research 23A**, 103-113.

Ripley, B.D. (1992) Statistical Aspects of Neural Networks. **Proc. SemStat**, Denmark.

Roberts, N.P., Webb, S.A. and Coe, G.A. (1994) Incidents on Motorways, **TRL Project Report 84**.

Roberts, S. and Tarassenko, L. (1994) A probabilistic resource allocating network for novelty detection. **Neural Computation**, Vol 6, 270-284.

Shepherd, S. and Chen, H. (1996) Investigation of network performance prediction: feasibility study - development of the test bed. **ITS Technical Note 402**.

Smith, B.L. and Demetsky, M.J. (1994) Short-term Traffic Flow Prediction: Neural Network Approach. **Transportation Research Record 1453**, 98-104.

Traven, H.G.C. (1991) A Neural Network Approach to Statistical Pattern Classification by "semiparametric" Estimation of Probability Density Functions. **IEEE Transactions on Neural Networks 2** (3), 366-377.