A computationally efficient method for online identification of traffic incidents and network equipment failures

Victoria J. Hodge[1], Rajesh Krishnan [2], Jim Austin[1] and John Polak[2]
[1]Department of Computer Science, University of York, UK
[2]Centre for Transport Studies, Imperial College London, UK

Abstract — **Despite the vast wealth of traffic data available, currently there is only limited integration, analysis and utilisation of data in the transport domain. Yet, accurate congestion and incident detection is vital for traffic network operators to allow them to mitigate the cost of traffic incidents. Recurrent (cyclical) traffic congestion tends to be managed using timetabled control measures or through the use of adaptive traffic control systems such as SCOOT and SCATS. However, for non-recurrent congestion with rapid onset, such as the congestion caused by a traffic incident or traffic equipment failure, traffic network operators have to quickly detect the problem and then determine the likely cause before selecting the most appropriate action to both manage the traffic network and mitigate the congestion. This is a complex task requiring specialist knowledge where assistance from automated tools will help facilitate the operator tasks. Automated detection is becoming an increasingly viable option due to the increased use of traffic sensors in the road network. Therefore, the aim of the FREEFLOW project is to provide an Intelligent Decision Support (IDS) tool which is designed to complement existing fixed-time traffic control systems and adaptive systems SCOOT and SCATS. IDS will use traffic sensor data to rapidly identify traffic problems, recommend appropriate interventions that worked in the past for similar problems and assist the traffic network operators to pinpoint the cause of the problem. Recommendations will be displayed to the network operator who will use this knowledge to select the most appropriate course of action. This paper describes and analyses the components of the IDS tool used for identifying incidents and faulty equipment.**

Index Terms — **Intelligent Decision Support, Traffic Management, Traffic State Estimation Modelling, Pattern Match, Incident Detection, Equipment Failure Detection**

I INTRODUCTION

In the UK, traffic network management generally involves manual monitoring and intervention implementation to supplement the timetabled or automated traffic control systems in place. For example, the UK motorway network is monitored and controlled by the National Traffic Control Centre (Highways Agency 2009) and the road network in local authorities such as London is monitored and controlled by the respective authority's staff such as the London Traffic Control Centre (Barton 2004). The network operators respond to traffic problems, determine the likely cause of the problem and then select the most appropriate action to take to both manage the traffic network and mitigate the congestion. Such intervention measures are often based on the operational experience of the person handling the problem.

A large amount of near-real-time and historic traffic data are available from various sensors and systems at any given Local Authority (LA). The aim of the FREEFLOW project (Glover et. al. 2008) is to develop tools and techniques to convert traffic data into intelligence to assist network managers, operators and also to aid the travelling public. The traffic management component of the work within FREEFLOW is called Intelligent Decision Support (IDS), which forms the focus of this paper. The full IDS tool will detect traffic problems, identify the likely cause and recommend suitable intervention most likely to mitigate congestion of that traffic problem. Previous papers analysed incident detection and intervention

recommendation (Krishnan et. al. 2010b). In this paper, we analyse incident detection and cause identification.

The rest of this paper is organised as follows. Section II provides an overview of the IDS functionality. Section II.A will present the state estimation, II.B will present the pattern-matching and II.C will present the spatial matching. The data and analyses are discussed in sections III and IV including the cause-suggestion functionality within the IDS. This is followed by discussion and conclusions in sections V and VI respectively.

II INTELLIGENT DECISION SUPPORT

The objective of the IDS described here is to (a) determine if there is a traffic problem using near-real-time data from traffic sensors and systems, and, if there is a problem, (b) identify the cause. The IDS is a knowledge-based system that uses information about past traffic incidents to identify the current incident and suggest the most likely cause. The IDS requires an historic database of traffic sensor data and traffic incident data for the application area. The IDS is designed to work online using near-real-time traffic data and large historic datasets. Hence, IDS needs to be computationally efficient.

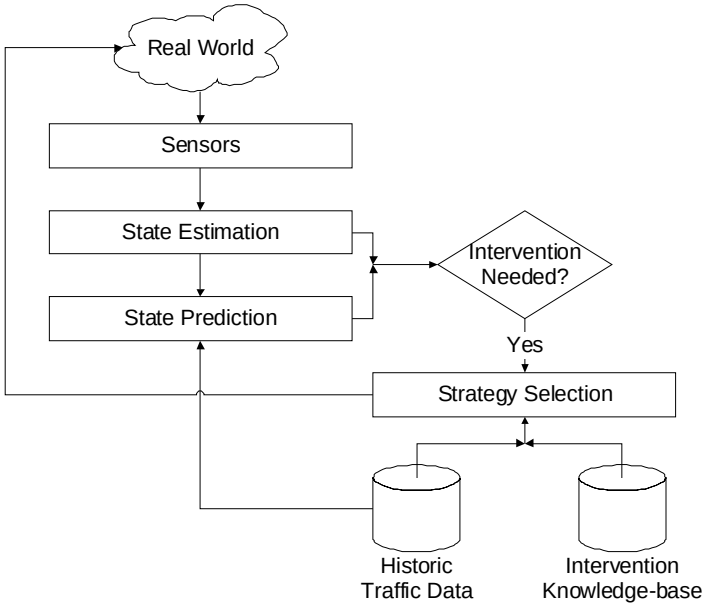

Figure 1: Logical overview of IDS

The IDS will monitor traffic sensor data to determine if the network is congested using traffic state estimation models developed at Imperial College London. The sensor data (typically flow and occupancy from Inductive Loop Detectors), is monitored at regular intervals (e.g. 5 minutes) over a geographical area of interest. The state identification algorithm is applied separately to each ILD (and thus each network link). The output of the state identification algorithm is binary: 0 if the link is uncongested and 1 if the link is congested. If one or more links are congested, the historic database is queried for similar congestion events using neural network based pattern matching tools developed at the University of York. The search consists of identifying the historical time periods when the traffic sensor data from the set of Inductive Loop Detectors (ILDs) is most similar to the currently observed data. Similarity is based on two components: magnitude and spatial similarity. Once the closest matches have been found, traffic incidents and equipment failures that occurred during similar congestion events in the historic database are then searched. The most similar historical case(s) will be displayed to the network operator along with an associated confidence indicator. A logical overview of the IDS system is given in Figure 1.

In this paper, the IDS is tested offline using Inductive Loop Detector (ILD) data obtained from the ASTRID system and incident and equipment failure log data obtained from the LTIS system at Transport for London (TfL). The paper presents the preliminary results using TfL data and outlines future research avenues for development.

A State estimation

A number of attempts to automatically determine the traffic state are available in the academic literature. Lao et al. (2004) use Fuzzy Logic to classify the traffic state into uncongested, "crowded" and congested using Fuzzy Logic; however, they used driver inputs rather than traffic sensor data. Narayanan et al. (2003) also used Fuzzy Logic to classify traffic using speed and inter-vehicle distance as input variables, using fixed thresholds in their classification method. Threshold based methods are generally not transferrable since the occupancy values reported by each ILD will depend on its electromagnetic sensitivity, and the thresholds could be different for different ILDs. Jiang et al. (2003) used Fuzzy Clustering of traffic sensor data consisting of flow, occupancy and spot-speed to cluster traffic into four states representing increasing levels of congestion. Of the above models, only the method presented in Jiang et al. (2003) provide a method to automatically identify the traffic state using traffic sensor data. However, the study does not provide a comprehensive evaluation of the proposed method. Moreover, the traffic states do not correspond to known traffic states in traffic engineering, though this criticism can be addressed by reducing the number of clusters in the proposed method. However, it is not clear if the modified method will correctly classify traffic into congested and uncongested states.

On the other hand, it is rather straightforward to visually classify traffic into congested and uncongested states using a scatter-plot of flow and occupancy values. Occupancy increases as the flow increases during the uncongested regime, and occupancy decreases as the flow increases during the congested state; see Figure 2 for illustration. However, it is not straightforward to develop an algorithm that can differentiate between the two traffic states. Direct application of a clustering algorithm, such as the k-means clustering (MacQueen 1967) method, leads to a number of congested data points being identified as uncongested. To address this problem, a two-step clustering approach was developed (Han et. al. 2009).

Step 1 clusters the data points into two clusters roughly representing congested and uncongested regimes using k-means clustering. The distance metric used is cosine, which uses the difference between the angles made by two different data points with the origin to determine cluster memberships. The use of the cosine distance metric takes advantage of the fact that the flow vs. occupancy curve is linear in the congested regime, and most of the uncongested data points should be grouped in the same cluster. However, due to the range of occupancy values in the congested regime, some of the congested data points may be classified into the first cluster of uncongested data points.

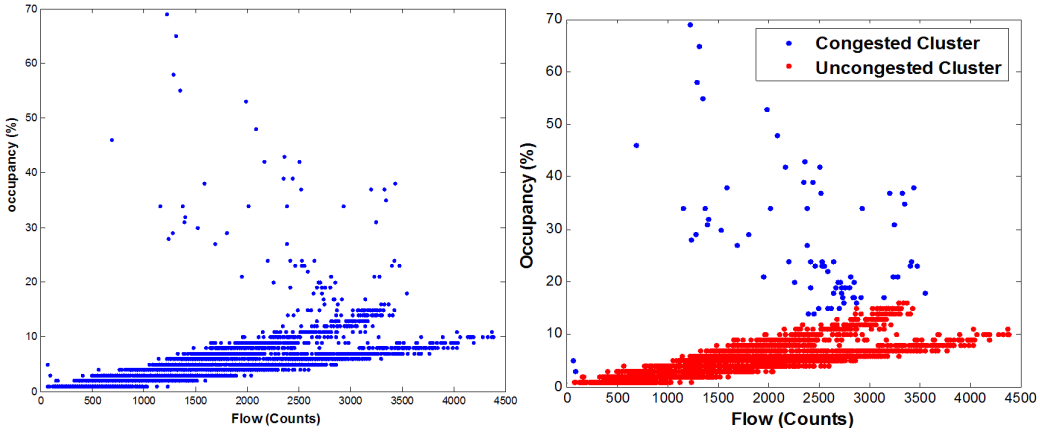

Figure 2: An example of the flow-occupancy plot of real ILD data

Step 2 fits a linear regression model on the data points in the uncongested cluster identified in the first step. All the data points identified as outliers by the regression model are moved to the second cluster, representing the congested state.

B Pattern Matching

The IDS pattern-matching module identifies time periods when the traffic state is most similar to the current observation. As with previous versions of the pattern matcher (Krishnan et. al. 2010b), we consider both magnitude and spatial similarity. The state is initially represented by a time-series of ILD readings and then incorporates spatial matching. The first stage of the pattern-matcher is a k-Nearest Neighbour (k-NN) technique implemented using the Advanced Uncertain Reasoning Architecture (AURA) technology (Hodge & Austin 2005) from the computer science discipline. Given the current vector X_q, and a historical dataset of vectors $\{X\}$, k-NN identifies the k nearest neighbours of X_q in $\{X\}$ using a distance metric. The commonly used distance metrics in k-NN are the Euclidean distance, unit map, Mahalanobis distance, city block distance and Minkowski distance. However, such metrics are insensitive to the position of the variables $\{x\}$ within the vector X_q; they calculate the magnitude of similarity but not the location of similarity and hence the location of congestion within the set of ILDs (Krishnan et. al. 2010). The concept of Centre of Mass (CM) is thus introduced in the second stage of pattern matching to address this problem.

The first step in pattern matching is to produce representations of the historical data and generate a fast-access data repository. For example, for two ILDs where ILD_1 has a vehicle count reading of 1 and occupancy reading of 5.0 and ILD_2 has vehicle count reading of 8 and occupancy readings of 45.0 then the historical vector X_h is:

$$X_h = \{1, 5.0, 8, 45.0\}$$

The AURA technology relies on binary searching for computational efficiency. The data vector X_h is converted to a binary string (I_h) using a process called quantisation (Hodge & Austin 2005). The quantisation process involves defining the range and precision of each variable in the data vector X_h, resulting in separate bins for different ranges of the variables within I_h. For example, for an integer-valued variable such as vehicle count per 5 minutes with range 0-9 and 5 bins then each bin would have width 2: bin 0 {0,1}, bin 1 {2,3} …bin 4 {8,9}. For a real-valued variable such as occupancy with range [0.0-100.0] and 5 bins then each bin would have width 20: bin 0 [0.0, 20.0), bin 1 [20.0, 40.0) …bin 4 [80.0, 100.0]. Thus the set of bin mappings for X_h are:

$$\text{Bins}(X_h) = \{0, 0, 4, 2\}$$

In the pattern matcher, each bin index maps to a binary representation so for five bins, bin 0 = 00001, bin 1 = 00010, bin 2 = 00100 etc. Thus, the bins corresponding to the values in X_h in are marked 1 while the other bins are marked 0. The binary representations for all the variables in the data vector are concatenated to create the binary string I_h which is a learning vector to allow the particular data vector to be stored and retrieved.

$$I_h = 00001\ 00001\ 10000\ 00100$$

The storage structure consisting of binary strings for all the observations in the data set $\{X\}$ is called a Correlation Matrix Memory (CMM) (Austin et. al. 1998). CMMs are the building blocks for AURA systems. AURA uses binary input and binary output vectors to train data into the CMM. Training is a one-pass process with one training step for each binary input string, i.e., each vector in the data set so training is rapid. Each binary string I_h is associated with a unique identifier vector O_h which has a single bit set to index a unique column in the CMM as given in Equation 1. This column thus indexes the binary string I_h.

$$\mathbf{CMM} = \bigvee_{\text{all h}} \mathbf{I}_h \times \mathbf{O}_h^T \quad \text{where} \quad \bigvee \text{ is the logical OR operator} \quad (1)$$

During retrieval, the CMM is searched to find the best matches. For each new query observation X_q, the retrieval vector R_q is created using a set of parabolic kernels, with one kernel for each variable x in X_q. The kernels may vary across variables according to the

number of bins assigned to that variable. In this paper all variables use an equivalent kernel. The kernel density is estimated using Equation 2.

$$Kernel(x_n) = \left[\left| \left(\frac{\max(b)}{2} \right) \right|^2 - \left(\left| bin(x_n^q) - bin(x_n^h) \right| \right)^2 \alpha(x_n) \right] \text{ where } \alpha(x_n) = \frac{(\max(b))^2}{(b(x_n))^2} \quad (2)$$

Where, $max(b)$ is the maximum number of bins across all variables, $|bin(x_n^q) - bin(x_n^h)|$ is the number of bins separating the bin mapped to by the variable value for the query vector (x_n^q) from the bin mapped to by the variable value for the stored historical vector (x_n^h), and $b(x_n)$ is the total number of bins for variable x_n.

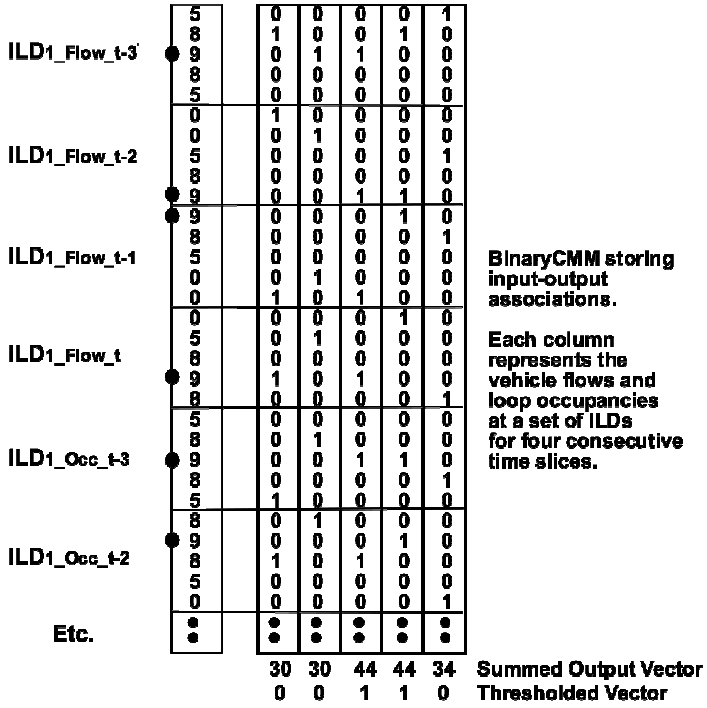

Figure 3: Illustrating the application of kernels to CMM to find the k-nearest neighbours using time-series vectors.

The columns of the matrix are summed according to the kernel weight on the rows indexed by the query retrieval vector R_q. The CMM produces a summed output vector S_q, as shown in Equation 3 and figure 3.

$$S_q^T = \sum R_q \bullet CMM \quad (3)$$

In AURA, the summed output vector S_q is thresholded using *L-Max* thresholding to produce a binary thresholded vector T_q. *L-Max* thresholding is used in the AURA k-NN as it retrieves the top L matches (Hodge & Austin 2005). After thresholding, T_q effectively lists the top L matching columns from the CMM thus identifying the top L matches. The AURA k-NN can perform up to four times faster than the traditional k-NN (Hodge & Austin 2005) thus allowing large data sets to be searched for the k nearest neighbours.

C Spatial Similarity

By incorporating the centres of mass (CM), the similarity between data vectors is calculated not only based on the distance but also based on the similarity of the spatial distribution of data values. Four different CM metrics were tested, along with original AURA matching, to find the best method for identifying similar incidents. The equation for calculating the CM is given in Equation 4.

$$CM = \frac{\sum m_i r_i}{\sum m_i} \quad (4)$$

Where m_i is the mass and r_i is the distance of object i from the origin.

However, the notion of distance and the mass in the context of binary input vectors are different for the four CM metrics. The CMM does not store the original data values but stores a quantised (binary) representation of each data vector. As the kernels used in the AURA k-NN utilise the quantisation bins to assign similarity, the proposed CM calculations also utilise the bins in an analogous manner for consistency and simplicity. The CM metrics used are given below where m_n is the mass for variable x_n, r_n is the distance for variable x_n and $\left|bin(x_n^q) - bin_0\right|$ is the number of bins between bin_0 and the bin representing the value for variable x_n of the query vector.

- CM-I: $m_n = \left| \quad bin(x_n^q) - bin_0 \quad \right|$ and r_a is calculated using the geo-coordinates (Easting and Northing) of the ILDs.
- CM-II: $m_n = (\max(b))^2 - \left(bin(x_n^q) - bin_0 \middle| \alpha(x_n)\right)^2$ and $r_n = \left| \quad bin(x_n^q) - bin_0 \quad \right|$
- CM-III: $m_n = (\max(b))^2 - \left(bin(x_n^q) - bin_0 \middle| \alpha(x_n)\right)^2$ and r_n = the ILD reading
- CM-IV: m_n = the ILD reading and r_n is calculated using the geo-coordinates (Easting and Northing) of the ILDs.

III TRAFFIC SENSOR DATA FROM LONDON

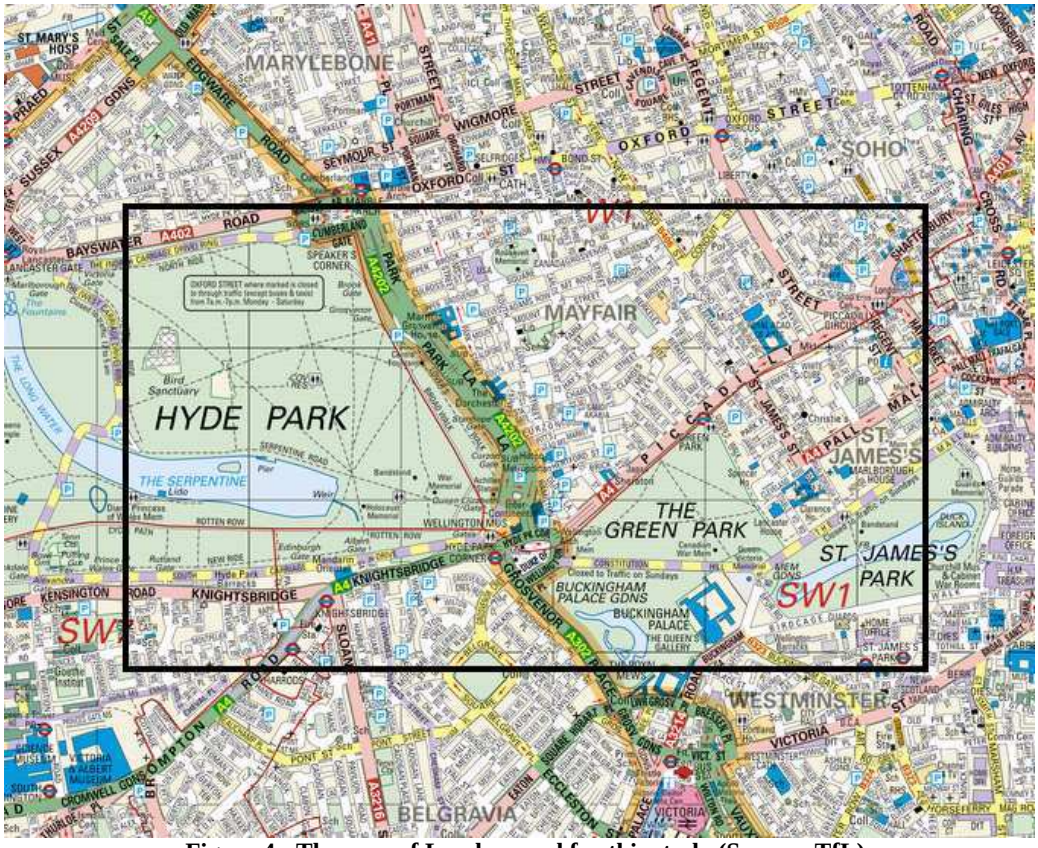

Figure 4: The area of London used for this study (Source: TfL)

Two datasets from Transport for London (TfL) were used to analyse IDS for recall and precision. SCOOT ILD data, consisting of flow and occupancy aggregated at 15-minute

intervals, was obtained from the ASTRID system. In addition, traffic incidents and equipment failures were obtained from the LTIS system as TfL. Both datasets covered a 12-month period from 1st Apr. 2008 to 31st Mar. 2009. The analyses use the area around Hyde Park Corner (HPC) comprising data from 32 ILDs in the area shown in Figure [4].

The objective of the analyses is to determine if the IDS can identify similar incidents and equipment failures in the historic data. Hence, three serious events in HPC area identified in the TfL's LTIS system (Barton 2004) were used for validation of the IDS pattern-matcher.

- Equipment fault on 14th May 2008
- Spillage on 15th May 2008
- Broken down vehicle 6th June 2008

IV RESULTS

The objective of the test is to determine how accurately the IDS method can identify time periods with similar congestion patterns. Given one time period within the duration of the event as input, it is expected that IDS should identify other time periods during the same event as time periods with similar congestion patterns. Moreover, IDS should identify other time periods when the congestion pattern was similar. In this section, the results consisting of top 5 matches and a qualitative analysis of the results are presented.

The ILDs are grouped together to form locations when determining the spatial accuracy of the match. For example, a given location on the road may have ILDs N01/381a and N01/381b on two separate lanes. Such ILDs are grouped together to form locations. The spatial accuracy of the match is determined based on the number of congested locations identified by the match.

Tables [1-3] show match statistics for the three incidents when the top 5 matches are retrieved by the various pattern match configurations. A "good" algorithm should identify time periods during the congestion build-up of the event or time periods during the duration of the incident. (The incident will be marked cleared only after the congestion due to the event dissipates). False positive (FP) values in the cells indicate the number of locations during the matched time period that are congested but not congested during the input time period. A higher value of FP means that the identified matches are congested at different locations. False negative (FN) values indicate the number of sensors that are congested during the input time period, but not during the matched time period. Table 4 shows aggregate results of the methods for all the incidents.

Matches	AURA		CM-I		CM-II		CM-III		CM-IV	
	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN
1	2	0	1	0	2	0	2	3	2	0
2	2	0	2	1	2	1	1	2	2	1
3	1	1	2	1	3	0	1	2	2	1
4	1	0	2	1	2	1	2	2	2	2
5	2	1	1	0	1	0	0	2	1	2
Total	**8**	**2**	**8**	**3**	**10**	**2**	**6**	**11**	**9**	**6**

Table 1: Results for equipment failure event on 14th May 2008

Matches	AURA		CM-I		CM-II		CM-III		CM-IV	
	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	1	1	2	2	1
3	1	0	0	1	1	0	0	1	0	1
4	0	1	0	1	0	1	0	1	1	1
5	0	1	2	1	0	1	0	1	0	1

Total	**1**	**2**	**2**	**4**	**1**	**3**	**1**	**5**	**3**	**4**

Table 2: Results for spillage event on 15th May 2008

Matches	AURA		CM-I		CM-II		CM-III		CM-IV	
	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN
1	0	4	0	2	0	2	1	1	0	3
2	0	3	0	2	0	1	1	1	0	2
3	0	2	0	3	0	3	0	5	0	4
4	1	1	0	2	0	3	1	1	0	2
5	0	2	1	1	1	1	0	4	0	3
Total	**1**	**12**	**1**	**10**	**1**	**10**	**3**	**12**	**0**	**14**

Table 3: Results for broken down vehicle event on 6th June 2008

Matches	AURA		CM-I		CM-II		CM-III		CM-IV	
	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN
Total	**10**	16	11	17	12	**15**	**10**	28	12	24

Table 4: Overall performance comparison of pattern-matching techniques

V DISCUSSION

FN indicates a more serious (from FREEFLOW perspective) problem of missed links in the match than FP. For example, if all the sensors that are congested during the current time period (input vector) and a few extra locations are congested during the matched time period (FP), the recommended intervention is still presumably valid as it is capable of easing congestion on the matched links. On the other hand, if some of the sensors that are currently congested are not congested in the matched time period (FN), potentially a different intervention should be used. Hence, a lower value of FN is more important than a lower value of FP.

Keeping these factors in mind, AURA, CM-I and CM-II perform the best. The use of spatial distance metrics actually makes the FP rate slightly worse than simple AURA but CM-II has the lowest FN rate which is the most important measure. The result with respect to FP rate was somewhat unexpected. An explanation may be that the relatively large number of sensors in the feature vector may obscure the spatial pattern of congestion. Selectively choosing the sensors for matching may solve this problem. However, preselecting ILDs is not simple as different ILDs are required to identify different congestion topologies. We would need to know the congestion topology to preselect the ILDs but the task is to identify (and recognise) congestion which is a circular cause and consequence. Hence, it is important that all the sensors in the area of interest are monitored since a potential problem could occur in any one of the links.

A The Use of Pattern-Matching In IDS

The matched time periods with similar congestion patterns form the input to the rest of the IDS algorithm. IDS may suggest potential causes of the incident by correlating the matched time periods with similar congestion patterns and incidents and equipment faults. This information may be displayed to the local authority staff and will provide a number of potential reasons behind the congestion event.

VI CONCLUSION

Two key components of the traffic management module developed within FREEFLOW are state estimation and pattern-matching. The spatial accuracy of matches using pattern-matching is critical to the accuracy of the proposed method. This paper describes different configurations of the AURA pattern-matcher and tested them against real ILD data from around Hyde Park Corner in central London. The test was carried out using data when

incidents were known to be present, and the accuracy of the pattern matcher was determined based on the number of congested locations identified by the matcher. It was recommended that the AURA k-NN and AURA k-NN in conjunction with both the CM-I and CM-II distance metric developed in the FREEFLOW project should be used for obtaining accurate location based congestion matches. The use of the output of the pattern-matcher to index historical cases and generate information for display to the network operator within the IDS traffic management module was also described in the paper.

ACKNOWLEDGEMENTS

The work reported in this paper forms part of the FREEFLOW project, which is supported by the UK Engineering and Physical Sciences Research Council, the UK Department for Transport and the UK Technology Strategy Board. The project consortium consists of partners including QinetiQ, Mindsheet, ACIS, Kizoom, Trakm8, City of York Council, Kent County Council and Transport for London.

REFERENCES

Austin, J., Kennedy, J. & Lees, K. 1998. The Advanced Uncertain Reasoning Architecture, AURA, In RAM-based Neural Networks, Ser. Progress in Neural Processing. World Scientific Publishing, 1998, vol. 9, pp. 43–50.

Barton, N. 2004. Keeping London Moving: Real Time Traffic Management Systems and Operations in Transport for London. Proceedings of the 32nd Annual European Transport Conference (ETC), Strasbourg, France. 4-6 October 2004.

Glover, P., Rooke, A. & Graham, A. 2008. Flow diagram, Thinking Highways, 3(3), pp. 20-23.

Han, J., Krishnan, R. and Polak, J. 2009. Traffic state identification using loop detector data. International Conference on Models and Technologies for Intelligent Transportation Systems, Sapienza University of Rome, Italy. June 2009.

Highways Agency, 2009. National Traffic Control Centre. Available from http://www.highways.gov.uk/knowledge/1298.aspx. Accessed on 15 Nov. 2009.

Hodge, VJ. and Austin J. 2005. A Binary Neural k-Nearest Neighbour Technique. Knowledge and Information Systems, 8(3), pp. 276–292.

Hunt, PB.. Robertson, DI.. Bretherton, RD. & Winton, RI. 1981. SCOOT - A traffic responsive method of coordinating signals. TRRL Laboratory Report 1014. Transport and Road Research Laboratory, UK.

Jiang, G., Wang, J., Zhang, X. and Gang, L. 2003. The study on the application of Fuzzy Clustering Analysis in the dynamic identification of road traffic state. IEEE 6th International Conference on Intelligent Transportation Systems, Shanghai, Oct. 12-15, 2003.

Krishnan, R., Hodge, V.J., Austin, J., Polak, J.W. & Lee, TC. 2010. On Identifying Spatial Traffic Patterns using Advanced Pattern Matching Techniques, 89th Annual Meeting of Transportation Research Board, Washington D.C., USA, Jan.10-14, 2010.

Krishnan, R., Hodge, V.J., Austin, J. & Polak, J.W. 2010. A Computationally Efficient Method for Online Identification of Traffic Control Intervention Measures. *42nd Annual UTSG Conference,* University of Plymouth, UK: Jan. 5-7, 2010

Lao, Y., Yun, M., Tang, S., Wang, C., Yang, X. and Chu, H. 2007. Evaluation Method for Accuracy of Road Traffic State Information. Proceedings of the First International Conference on Transportation Engineering 2007 (ICTE 2007). Chengdu, China: Jul. 22-24, 2007.

MacQueen, JB. 1967. Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press. pp. 281–297.

Narayanan, R., Udayakumar, R., Kumar, K. and Subbaraj, L. 2003. Quantification of congestion using Fuzzy Logic and Network Analysis using GIS. Proceedings of Map India Conference 2003. New Delhi, India: Jan. 28-31, 2003.