Pergamon

PII: S0968-090X(96)00015-0

COMBINING KOHONEN MAPS WITH ARIMA TIME SERIES MODELS TO FORECAST TRAFFIC FLOW

MASCHA VAN DER VOORT,[1] MARK DOUGHERTY[2] and SUSAN WATSON[3]

[1]Department of Civil Engineering and Management, University of Twente, Enschede, The Netherlands
[2]Centre for Research on Transportation and Society, Darlana University, Darlana, Sweden
[3]Institute for Transport Studies, University of Leeds, Leeds, U.K.

Abstract—A hybrid method of short-term traffic forecasting is introduced; the KARIMA method. The technique uses a Kohonen self-organizing map as an initial classifier; each class has an individually tuned ARIMA model associated with it. Using a Kohonen map which is hexagonal in layout eases the problem of defining the classes. The explicit separation of the tasks of classification and functional approximation greatly improves forecasting performance compared to either a single ARIMA model or a backpropagation neural network. The model is demonstrated by producing forecasts of traffic flow, at horizons of half an hour and an hour, for a French motorway. Performance is similar to that exhibited by other layered models, but the number of classes needed is much smaller (typically between two and four). Because the number of classes is small, it is concluded that the algorithm could be easily retrained in order to track long-term changes in traffic flow and should also prove to be readily transferrable. Copyright © 1996 Elsevier Science Ltd

1. INTRODUCTION

Short-term forecasting of traffic flows is an essential part of traffic control and information systems. With a good estimate of what the flow will be in half an hour or an hour, it is possible to take action now. By taking the right action in advance, it is then possible to avoid or at least reduce congestion. A considerable amount of effort has therefore been expended on this problem, with a variety of different algorithms proposed. These include Kalman filters; Okutani and Stephanedes (1984) and spectral analysis; Nicholoson and Swann (1974) as well as the techniques mentioned below.

The work described in this paper was inspired by two previous comparative studies of three different traffic flow forecasting techniqes; Clark *et al.* (1993) and Kirby *et al.* (1994). The techniques concerned were back propagation neural networks, Box-Jenkins ARIMA models and a method known as ATHENA (Danech-Pajouh and Aron, 1991). This third method uses a layered statistical approach, with a mathematical clustering technique to group the data and a separately tuned linear regression model for each cluster.

The results of the studies showed that ATHENA was superior to the other two methods, particularly as the forecasting horizon was extended further into the future. However, the ATHENA method is very complicated, with up to 192 different clusters needed for just a single geographical forecasting point. It can therefore be regarded as a 'brute force' approach, with a vast number of parameters which need to be tuned. Such approaches are often effective, but suffer from practical difficulties. For example, retraining in response to changes of the physical system or transfer to another site can be a laborious process. On a more philosophical level, a brute force approach will not greatly help our overall understanding of the problem and in this context a more generalized solution is preferable.

However, the ATHENA method clearly showed the advantages of a layered model, and its numerical performance was unarguably superior to back propagation networks and ARIMA models. It was therefore decided to experiment with models similar in

structure to the ATHENA method, but with different sub-components. In particular, we wanted to explore the possibility of reducing the number of clusters needed by using more complex forecasting modules. In this way, we hoped to create a more generalized method with a better balance between the two layers.

2. THE DATA

The data used in all cases were the same as that used during the ATHENA project, originating from detectors at four French motorway sites. The principal site was at Beaune, where the flow along three feeder motorways converges onto a single motorway. Each feeder motorway had a single detector site approx. 30 km upstream of the convergence point. The three upstream measurement points were at Avallon, Baume-les-Dames and Langres (Fig. 1).

The historical flow data were aggregated across the entire carriageway and averaged over half hour periods. Predictions of the flow at Beaune were made for horizons of half an hour and one hour into the future. For forecasting the flow an hour ahead the data were aggregated to hourly data by summing two half hourly data points. Data for the months of July and August from 1984 to 1989 were used for training the models. A test data set was also available containing data from July and August 1990. Note that this was only used at the very end of the research to validate the best models produced.

The rationale for the chosen format of the data is not justified in this paper as in order to make an accurate comparison with the ATHENA work we were naturally constrained to use the same general formulation of the problem.

3. CLUSTERING THE DATA SET BY WEEKENDS AND WEEKDAYS

As mentioned in Section 2, classical ARIMA(p,d,q) models were used to compute the forecasts, where.

p is the number of auto-regression terms
d is the number of difference terms
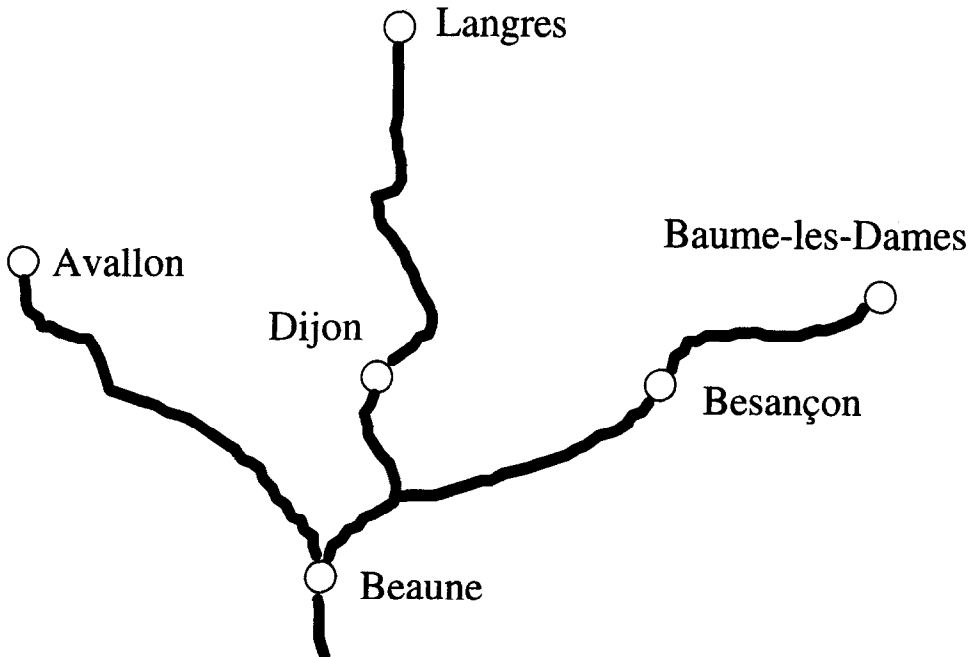q is the number of moving-average terms

Fig. 1. Geographical relationship of data collection points.

In this case, the models contained no difference terms. The general equation of an ARIMA(*p,d,q*) model is the following:

$$\hat{y}_t = \Phi_1 \ y_{t-1} + \Phi_2 \ y_{t-2}\dots\dots + \Phi_p \ y_{t-p} + a_t - \theta_1 a_{t-1}\dots\dots - \theta_p a_{t-q}$$

where:

y	denotes a general time series
$\hat{y}_t$	is the forecast of the time series y for time t
$y_{t-1}\dots y_{t-p}$	are the previous p values of the time series y (and form the auto-regression terms)
$\Phi_1\dots\Phi_p$	are coefficients (to be determined by fitting the model)
$a_t\dots a_{t-q}$	is a zero mean white noise process (and forms the moving average terms)
$\theta_1\dots\theta_{t-q}$	are coefficients (to be determined by fitting the model)

This equation can be easily extended into a multivariate one by expanding Φ, θ and y into vectors of coefficients and variables respectively. In the context of the problem domain this is done by including data from the three upstream points as input variables of the model. For all the models discussed in this paper the multivariate procedure was used. For a full coverage of ARIMA models the reader is referred to Box and Jenkins (1976).

Clustering was done manually using chronological factors. Initially we divided the data set into the months July and August. The rationale behind this was that August is the main holiday season in France, and is therefore likely to have different traffic flow patterns compared with other months. As we suspected that the pattern of the flow differs between weekends and weekdays, we also divided the data set into weekends and weekdays. For each data set a statistical forecasting model was developed to see if the overall performance of the two separate models was an improvement over a single model for the whole week.

Initially, twelve new data sets were made from the 1984–1989 data, using all possible combinations of the following three options:

- Forecast horizon — half an hour or one hour
- Month — July or August
- Days of week — whole week, just weekdays, just weekends

To avoid edge effects, the last five halfhourly or hourly measurements of the flow on Friday were used as boot-strap inputs to the data set of the weekends. The same principle was used for the weekdays data set. Here the last five halfhourly or hourly values of the flow on Sunday were added to the data set.

For each data set the results of many differently configured ARIMA models were compared. A range of values of both the auto-regression parameter (p) and the moving-average parameter (q) were experimented with. To select the best model for one particular data set, diagnostics such as Akaike's information criteria and the auto correlation function were used. In general it was found that the model with the best results for a data set had a value for p between 1 and 4 and a value for q of 0 or 2. The SAS software package was used to estimate the model parameters.

Results from the best models created for each data set are shown as an error distribution. To allow for easier comparisons, in the case of the seperate models for weekdays/weekends, the error distributions are combined into a single distribution. The results are given in Table 1.

From these error distributions it can be concluded that predicting the flow by using two separate models for weekdays and weekends shows little or no improvement over the results of the models for the whole week. It is also clear that it is harder to achieve good forecasting performance for August; this is not surprising as during the holiday season there tends to be more extreme congestion due to incidents.

To give a further comparative view, it is interesting to plot some of these results as a histogram alongside results from the ATHENA model. To do this we combine results of

Table 1. Error distribution for manual clustering using temporal factors

	< −25	−25 to −15	% rel. error −15 to −5	−5 to 5	5 to 15	15 to 25	> 25
Hourly July 1984-89							
Week	2.09	5.78	25.55	42.93	20.22	3.01	0.38
Weekends + weekdays	2.61	7.68	25.26	40.58	20.67	3.52	0.79
Hourly August 1984-89							
Week	3.89	9.33	23.44	38.52	20.90	3.26	0.62
Weekends + weekdays	3.63	9.281	23.66	39.61	19.44	3.54	0.81
Half-hourly July 1984-89							
Week	1.72	5.71	23.18	44.59	21.72	2.60	0.47
Weekends + weekdays	2.57	6.42	23.33	42.88	21.60	2.59	0.57
Half-hourly August 1984-89							
Week	3.67	7.93	23.21	39.09	22.15	3.47	0.48
Weekends + weekdays	3.74	8.15	23.17	38.95	21.69	3.74	0.53

the hourly forecasts for July and August into a single overall distribution in a similar fashion to combining the weekday/weekend results (Fig. 2).

It is plainly evident that the ATHENA method is superior. One could possibly conclude that we failed to use enough clusters, and that performance could be improved by further subdivisions such as peak/off-peak and so on. However, to substantially increase the number of clusters would be heading towards a 'brute force' approach, and thus be little different from the ATHENA method. We therefore concluded that in order to achieve our objective what was needed was not more clusters, but a better way of clustering than the simplistic method used thus far.

4. NEURAL NETWORKS

4.1. Previous work using neural networks

Neural networks have been proposed by several groups as a possible method for short-term forecasting of traffic; Dougherty *et al.* (1993), Dougherty and Cobbett (1994),
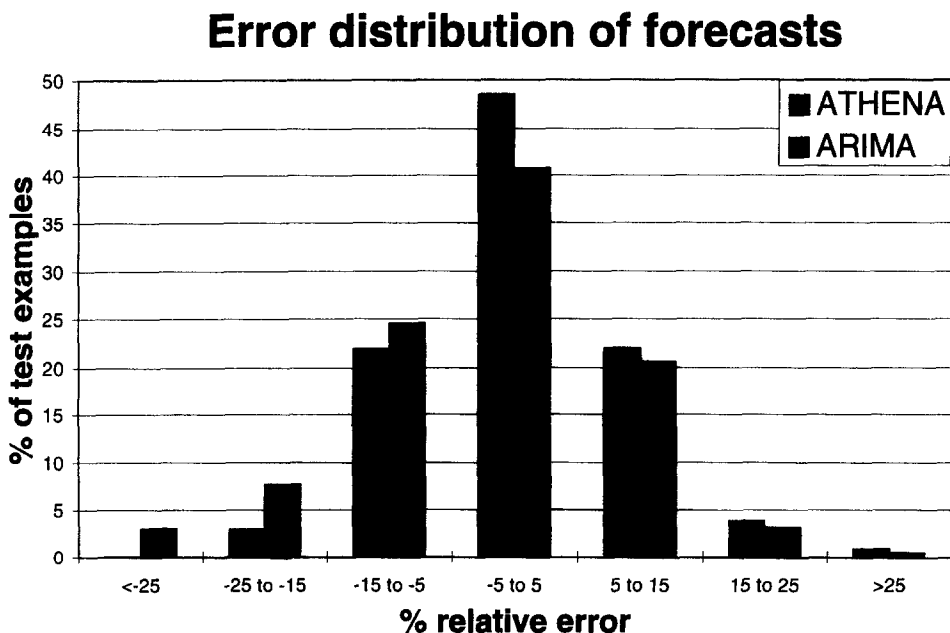
Error distribution of forecasts

Fig. 2. Comparison of ATHENA and ARIMA.

Smith and Demetsky (1994) and Dochy *et al.* (1995). The usual approach taken has been to use multilayer feedforward networks (also known generically as back propagation networks) to perfom a generalized mapping between flows from the recent past and future flows. It is clear that this approach is only moderately successful; some inprovements over statistical techniques such as linear regression and ARIMA are noted, but performance is still generally reported as worse than the ATHENA method (Kirby *et al.*, 1994). An exception to this general rule is work reported in Dochy *et al.* (1995), using the Beaune data set once more but applying back propagation networks. Performance is very competitive with ATHENA. However, this work procedes along the same 'brute force' lines as the ATHENA method, by using a large number of individually tuned neural networks for different times of the day, and is also likely to suffer from the problems outlined above.

Instead of being rivals when searching for the best method to forecast traffic flow, we propose an approach where neural networks and statistical models work together. Thus within the layered structure already dicussed, a neural network is used to cluster the data and for each cluster formed by the neural network, a statistical ARIMA model is then developed.

For comprehensive coverage of the subject of neural networks, the reader is referred to Hassoun (1995).

4.2. Kohonen networks

The type of neural net chosen to cluster the data was a Kohonen self-organizing map (Kohonen, 1995). Use of this paradigm has so far been quite rare within the transport sector (Dougherty, 1995). There were three reasons for this choice:

- Reinforcement learning is very suited to classification problems.
- We needed an unsupervised network.
- Kohonen maps provide good visual feedback to the user.

Self organizing maps have just two layers; a linear input layer which is then wholly connected to the map itself via weighted connections (Fig. 3). As a vector of input data is presented, each of the neurons in the map is stimulated. The neuron with the highest activity is awarded the status of 'winner' and its connection weights are increased. So far
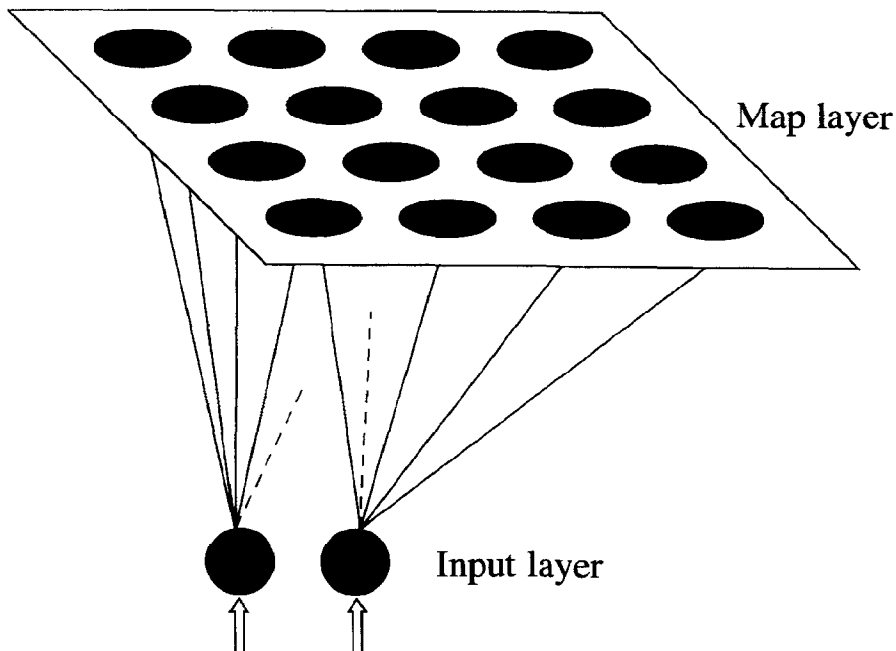

Fig. 3. Structure of a Kohonen map.

this seems little different from many other paradigms. However, what makes the SOM paradigm different from most other paradigms is that the spatial relationship of the neuron in the map is important. When a neuron is declared a 'winner', neurons within a certain radius of influence, the 'neighbourhood function', of that neuron also have their connection weights increased (though by less than the winner). This technique encourages groupings of highly active neurons to develop, and one finds that input data vectors which stimulate those neurons within the same spatial grouping tend to be similar in some fashion. Thus by interpreting the groupings observed on the map, clustering of the data can be carried out.

There are many variations on this general scheme. Neural network practitioners who are unfamiliar with the SOM paradigm will recognize within the literature principles used in the optimization of other types of network. For example, one can begin training a network with a large area of influence and gradually reduce it in size. This obviously requires a dynamic neighbourhood function. One can also extend the learning scheme to actively reduce the weights of those neurons lying at the edge of the area of influence to try and sharpen the edges of groupings; this is known as a 'Mexican hat' neighbourhood function. One can also experiment with maps of many different sizes and layouts; one is not restricted to a two-dimensional map. However, two basic 2-D topologies are the most popular configurations (Fig. 4):

- a rectangular lattice of neurons (each neuron has 4 nearest neighbours)
- a hexagonal lattice of neurons (each neuron has 6 nearest neighbours)

5. COMBINING SQUARE KOHONEN MAPS AND ARIMA MODELS

Initial experiments were carried out using the software package NeuralWorks Professional II Plus. This only supports self-organizing maps which have a rectangular grid layout of neurons. In these trials the self-organizing map had dimensions of 20×20 nodes.

An initial trial data set containing vectors of 16 variables was made and used as the input for the self-organizing map. Forecasts were only made for a half-hour horizon. Each vector contained the following items of data:

- the most recent half-hour flow measurement at the four sites ($f_{t=0}$)
- the difference in flow from the previous values of flow ($f_{t=0} - f_{t=-1}$)
- a time variable to indicate the half hour (ranging from 0 to 47)
- seven binary variables, each indicating a different day of the week

The trained map exhibited seven very strong clusters (Fig. 5). Examination of the data showed that vectors from each day of the week were simulating a different cluster. We concluded that the binary variable of the day of the week was dominant as an input for the map. As we were hoping to find a more sophisticated clustering criteria, a new data set was created with the binary variables indicating day of the week omitted (leaving vectors of 9 variables). The map produced with this data set showed two large clusters and

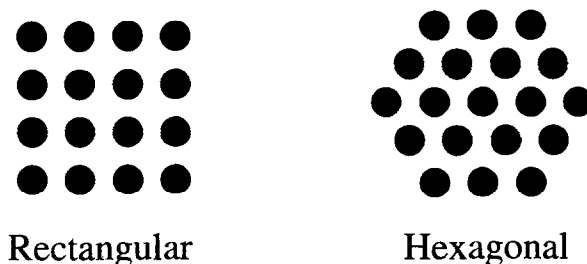
Rectangular   Hexagonal
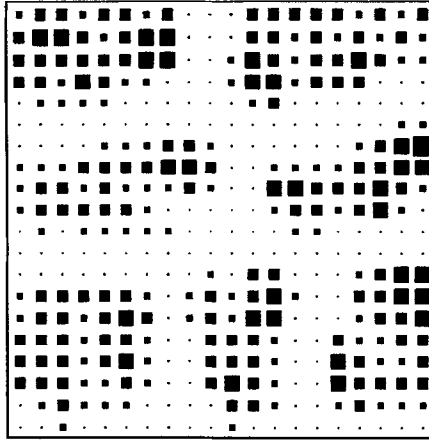
Fig. 4. Alternative map topologies.

Fig. 5. Clustering dominated by day-of-week variable.

one smaller cluster (Fig. 6). This outcome did not obviously mirror any particular input variable and so the clustering was potentially of some worth.

Initial attempts at interpretation were done by simply splitting the map up into 16 equal squares. A separate ARIMA model was trained for each of the 16 squares. Once again different configurations of ARIMA were tried for each forecasting module. Twelve ARIMA(p,0,q) models with values for p between 1 and 5 and with values for q between 1 and 3 were considered. The best models were determined using the same diagnostics previously described.

The performance of some of the models proved to be good, others poor. To determine the performance of the model as a whole, the error distribution of each model was cumulated to give an error distribution for the whole data set. This error distribution is listed in Table 2.

When these results are compared with the results of the models using data sets that were divided into weekdays and weekends (see Table 1), it can be seen that the performance is inferior.

Because the results of this experiment were less impressive than hoped for, another approach was tried. The total data set was divided in three separate data sets; one containing all the lines of data with coordinates that are highly active on the map, one data set containing all the lines of data with coordinates that are moderately active and one containing all the lines of data with coordinates which were less active on the map. Once again, for each data set ARIMA(p,0,q) models were made.
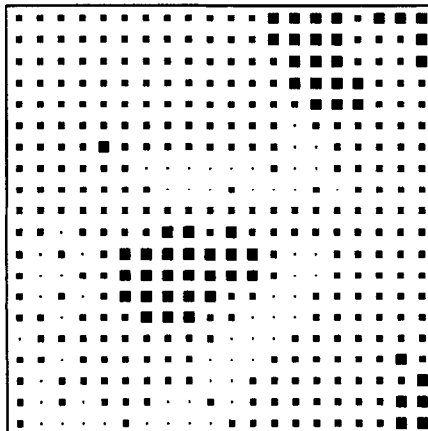
Fig. 6. Final attempt using square Kohonen map.

Table 2. Error distribution when clustering using 16 sub-squares

% rel. error	Half-hourly July and August 1984–89						
	< -25	-25 to -15	-15 to -5	-5 to 5	5 to 15	15 to 25	> 25
Overall	4.48	7.92	21.89	35.95	21.37	6.45	2.1

Table 3. Error distribution when clustering by strength of activity

% rel. error	Half-hourly July and August 1984–89						
	< -25	-25 to -15	-15 to -5	-5 to 5	5 to 15	15 to 25	> 25
High	10.97	10.97	20.56	29.85	17.03	6.67	3.94
Med.	6.65	9.94	22.40	32.76	19.71	5.66	2.88
Low	7.12	9.24	20.83	33.45	21.92	5.16	2.29
Overall	7.67	10.08	21.79	32.19	19.37	5.82	3.04

The error distribution (Table 3), shows that the total performance of these models is even worse than the sub-square clustering.

It was concluded that the performance of models produced for data selected using 16 equal sized grids was better than that for data separated by level of activity. However, the results using both criteria were still poor, and so forecasts of an horizon of an hour were not attempted. We concluded that what was needed was to manually identify clusters so as to make use of the spatial relationships which are the *raison d'être* of using the self-organizing map in the first place. However this was not possible because of the very large number of neurons all recording a medium level of activity. These neurons account for a substantial percentage of 'winners', but there seems no way of improving forecasting performance for this group.

6. COMBINING HEXAGONAL KOHONEN MAPS AND ARIMA MODELS

To improve the performance, hexagonal maps were tried as an alternative to a square grid. As has been reported, one of the main advantages of hexagonal maps is easier interpretation (Kohonen, 1995), it was hoped that the problem of how the clusters should be defined could be solved.

Because (at the time of writing) NeuralWorks did not support hexagonal maps, an alternative software package was located; SOM_PAK. This is freely available from the Helsinki University of Technology. An interesting feature of this software is that it calculates a weighted quantization error at each training iteration. Thus one can see the convergence of the network even though the learning is unsupervised.

The self-organizing maps used had 15 rows of 20 nodes each. Each row is offset from the previous row so as to give a hexagonal structure. This is topologically similar to the concept of offsetting alternate rows of seats in a cinema.

Once again we began with a half-hour forecast horizon, and experimented with various different combinations of input variables. The data set which showed most promise with the hexagonal map contained the half-hour time variable, the halfhourly flow on the four sites and the seven binary variables indicating the day of the week. Unlike the square map, the hexagonal map did not converge on the simplistc solution of producing a cluster for each day of the week (Fig. 7). Note that on this figure all the neurons are the same size; activity is indicated by the shade of the neurons. A dark shade indicates high activity, a light shade low activity.

The data sets were now divided into two according to the amount of activity (high or low) on the map. Several ARIMA$(p,0,q)$-models were once again made with values for p
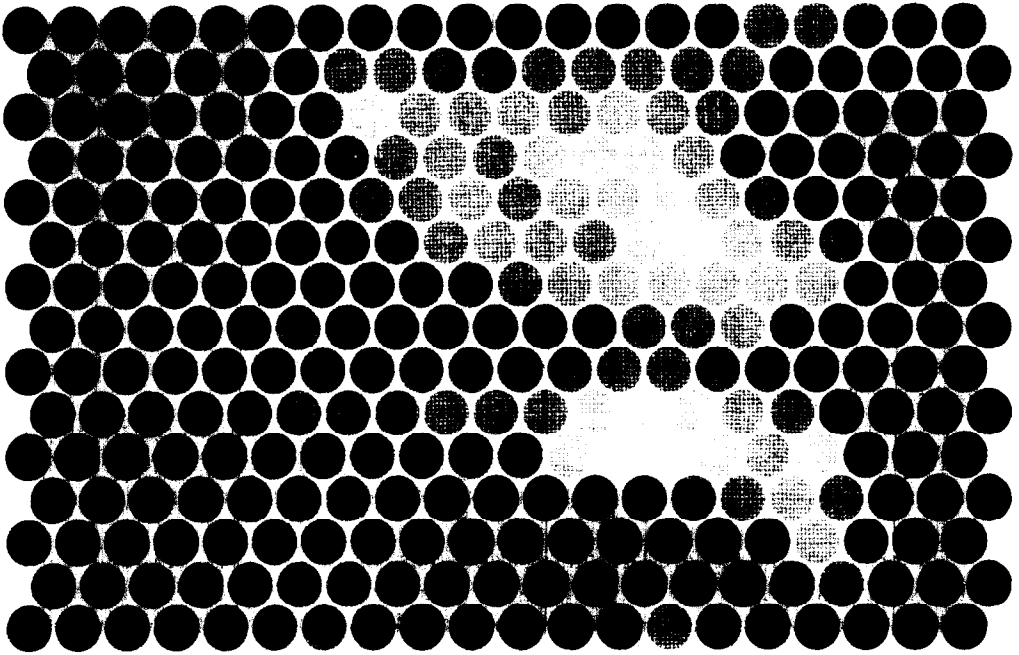
Fig. 7. Hexagonal Kohonen map.

between 1 and 5 and for q between 1 and 3 were tried on these two data sets. In this case it was found that the best models had a value of 2 or 3 for p and a value of 1 or 2 for q. Results for this binary clustering procedure are shown in Table 4.

Progress was now apparent — there was a marked improvement in performance of the model, particularly with repsect to the model tuned to data associated with the less activated neurons. This was unexpected as no spatial information had been used to define the clusters at this stage.

Because the error distribution showed that the performance of the 'highly activated' model was worse than the performance of the 'less activated' model, the data set containing the data with highly activated coordinates was further divided into three different data sets. Each of these data sets corresponds approximately with a spatial cluster on the map. These were defined manually by examining the map, there was therefore some level of subjectivity and trial and error in defining the clusters. Using the same method as above the best ARIMA-model for each data set was found.

The error distributions of these three separated models are shown in Table 5. Note that the results are then recombined into a single distribution (which can be compared with the 'highly activated' model results in Table 4). This is then further combined with the 'less activated' model results in Table 4 to produce an overall error distribution for this modified method.

These results show that the performance of the three separated models is slightly better than the performance of the model for the whole data set containing data corresponding

Table 4. Error distribution for binary clustering by hexagonal Kohonen map

% rel. error	July and August 1984–89						
	< -25	-25 to -15	-15 to -5	-5 to 5	5 to 15	15 to 25	> 25
High	5.98	9.43	21.23	34.23	21.02	6.06	2.00
Less	1.63	4.89	22.08	49.62	17.62	3.47	0.68
Overall	3.01	6.33	21.81	44.69	18.70	4.30	1.10

Table 5. Error distribution when highly activated points are spatially divided

% rel. error		July and August 1984–89					
	< −25	−25 to −15	−15 to −5	−5 to 5	5 to 15	15 to 25	> 25
High 1	6.03	11.28	22.66	29.91	20.57	6.99	2.51
High 2	0.90	5.7	25.22	42.04	24.02	2.1	0.00
High 3	5.00	6.34	19.02	40.77	23.66	4.58	0.58
High combined	5.34	9.09	21.49	34.71	21.94	5.78	1.63
Overall	2.88	6.23	21.89	44.84	18.99	4.21	0.99

with highly activated points. It was therefore considered worthwhile validating this model with the test data from 1990.

The test data set was therefore clustered in exactly the same way as the training set. The ARIMA models used were those found to be best for the training set and in fact some extra analysis showed that other configurations of ARIMA model did not fit the data as well in any case.

The results of the test are laid out in Table 6 as follows. H1, H2, H3 represent the data sets containing data corresponding to highly activated clusters of neurons, whilst L represents the data set containing the data corresponding to the less activated neurons.

It is surprising to note that these results are slightly better than the model tested on the training data. This is a very satisfactory demonstration that the method generalizes well.

Because the results were so promising the performance of a model for the hourly flow was also explored using exactly the same methodology. The only difference was that a sub-division of data associated with highly active coordinates into more was not relevant as the Kohonen map revealed only one cluster. Results for the test set are shown in Table 7.

As the error distribution shows, the results of this hourly forecasting model are in fact marginally better than the half hour forecasts made by the same method!

At this point it is interesting to return to the results from the ATHENA method. Figure 8 shows a histogram of ATHENA results side by side with the Kohonen enhanced ARIMA results.

Overall performance is very similar. However, the ATHENA model does exhibit a more symmetric distribution in the error structure. In contrast, the Kohonen enhanced ARIMA model shows considerably more under-estimates than over-estimates.

Table 6. Error distribution of the test data set (half-hour horizon)

% rel. error		Half-hourly July and August 1990					
	< −25	−25 to −15	−15 to −5	−5 to 5	5 to 15	15 to 25	> 25
H1	1.90	3.818	21.72	48.50	21.61	2.07	0.38
H2	0.00	3.79	16.45	64.55	13.92	1.26	0
H3	0.00	1.48	1.48	33.33	37.03	0	0
L	3.17	5.96	22.46	42.38	20.08	4.31	1.13
Overall	2.18	4.54	21.71	47.05	21.27	2.07	0.55

Table 7. Error distribution for the test data set (one hour horizon)

% rel. error		Hourly July and August 1990					
	< −25	−25 to −15	−15 to −5	−55 to 5	5 to 15	15 to 25	> 25
High	1.08	3.837	22.88	50.63	19.04	2.21	0.35
Less	2.77	6.34	24.75	41.78	19.00	4.75	0.58
Overall	1.68	4.76	23.57	47.36	19.03	3.15	0.44

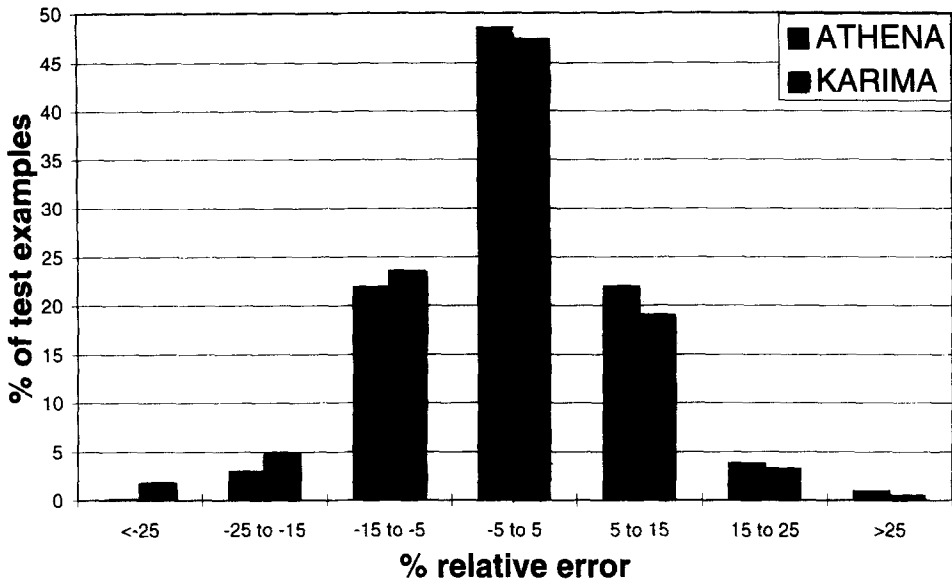# Error distribution of forecasts

Fig. 8. Comparison of ATHENA and KARIMA.

7. CONCLUSIONS

A level of numerical performance equal to the ATHENA method was reached. However, it must be pointed out that the model was only tested on two months of data. This is not enough to explore long-term robustness. Given that the model does not modify its coefficients on-line or contain a difference term, this means that it cannot cope with non-stationarities and thus will inevitably lose performance over a long period of time as conditions change. This problem affects many of the forecasting methods mentioned in this paper, including all three of the layered models which seem very attractive from the point of view of their performance, Kohonen enhanced ARIMA, ATHENA and the method explored in Dochy et al. (1995). However, the Kohonen enhanced ARIMA method it does have advantages compared with the other layered models with respect to this retraining problem. These come about because the Kohonen map can be retrained automatically (since it is a self-learning structure) and the number of classes is very small, meaning that only a small amount of effort is needed to re-fit the ARIMA models. We conclude that the approach of combining Kohonen maps and ARIMA models is potentially very useful, and have coined the expression KARIMA to refer to such a model.

The data set we had did not facilitate a study into the question of transferability. When building a new model for a different geographical site one must (in addition to training the model) specify the input vector for the Kohonen map and the values of p and q for the ARIMA models. This may take some manual effort. However, the number of ARIMA models is small, and in practice we found that there was little variation in the optimum values of p and q. These points argue in favour of the KARIMA method proving to be easily transferable. However, whether our chosen input vector for the Kohonen map is generally applicable is open to conjecture and this could prove a weakness. It is hoped that future work will investigate this point.

The hexagonal Kohonen map was found to be superior to a rectangular layout, although we can offer no theoretical explanation as to why this should be so. Whilst our initial reason for using a hexagonal layout was for easier manual interpretation, we found that the simplistic scheme of clustering by activity level (with no use made of spatial information) improved performance considerably (see Table 4). This result was unexpected, and was not the case with rectangular layout (see Table 3), where this approach actually produced worse performance than no clustering at all.

It was not possible to completely automate the process of fitting the very best model produced for the half hour forecast, as a manual interpretation of the clusters was found to improve performance. However, many suitable algorithms for interpreting objects similar to Kohonen maps (pixel maps) have been developed within the world of image processing. An interesting possibility for the future is therefore to incorporate this additional piece of technology in order to solve the difficulty.

Apart from the KARIMA method, the other two layered models mentioned in this paper also operate by splitting the data into several classes and fitting a separate model to each class. The performance of all three models is substantially better than other techniques such as straightforward ARIMA models and back propagation networks. It is concluded that traffic flow data exhibits a very high degree of non-linearity; the 'divide-and-conquer' scheme of the layered models seems well suited to overcoming this difficulty. They do so in a manner which is easy to understand and construct, since the component parts are relatively simple.

REFERENCES

Box, G. E. P. and Jenkins, G. M. (1976) *Time Series Analysis, Forecasting and Control*, 2nd edn. Holden-Day, San Francisco.
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. *Proceedings of the PTRC Summer Meeting*, Manchester, September 1993.
Danech-Pajouh, M. and Aron, M. (1991) *ATHENA, A Method for Short-term Inter-urban Traffic Forecasting.* English version of Report 177, INRETS, Arcueil, Paris.
Dochy, T., Danech-Pajouh, M. and Lechevallier, Y. (1995) Short-term road forecasting using neural network. *Recherche Transports Securite (English Issue)*, 11, 73–82.
Dougherty, M. S. (1995) A review of neural networks applied to transport. *Transportation Research C*, 3(4), 247–260.
Dougherty, M. S. and Cobbett, M. R. (1994) Short term inter urban traffic forecasts using neural networks. *Proceedings of the Second DRIVE—II Workshop on Short-Term Traffic Forecasting.* Delft, December 1994. (Currently in press with the *International Journal of Forecasting*).
Dougherty, M. S., Kirby, H. R. and Boyle, R. D. (1993) The use of neural networks to recognise and predict traffic congestion. *Traffic Engineering and Control*, 34(6), 311–318.
Hassoun, M. H. (1995) *Fundamentals of Artifical Neural Networks.* MIT Press, Massachusetts.
Kirby, H. R., Dougherty, M. S. and Watson, S. M. (1994) Should we use neural networks or statistical models for short term motorway traffic forecasting? *Proceedings of the Second DRIVE–II Workshop on Short-Term Traffic Forecasting.* Delft, December 1994. (Currently in press with the *International Journal of Forecasting*).
Kohonen, T. (1995) *Self Organising Maps.* Springer-Verlag, Berlin.
Nicholoson, H. and Swann, C. D. (1974) The prediction of traffic flow volumes based on spectral analysis. *Transportation Research*, 8, 533–538.
Okutani, I. and Stephanedes, Y. J. (1984) Dynamic prediction of traffic volume through Kalman filtering theory. *Transportation Research B*, 18(1), 1–11.
Smith, B. L. and Demetsky, M. J. (1994) Short-term traffic flow prediction: neural network approach. *Transportation Research Record*, 1453, 98–104.