2004 IEEE Intelligent Vehicles Symposium
University of Parma
Parma, Italy • June 14-17, 2004

A Comparison Of The Performance Of Artificial Neural Networks And Support Vector Machines For The Prediction Of Traffic Speed

Lelitha Vanajakshi, and Laurence R. Rilett, Ph. D., P. E.

Abstract— **The ability to predict traffic variables such as speed, travel time or flow, based on real time data and historic data, collected by various systems in transportation networks, is vital to the intelligent transportation systems (ITS) components such as in-vehicle route guidance systems (RGS), advanced traveler information systems (ATIS), and advanced traffic management systems (ATMS). In the context of prediction methodologies, different time series, and artificial neural networks (ANN) models have been developed in addition to the historic and real time approach. The present paper proposes the application of a recently developed pattern classification and regression technique called support vector machines (SVM) for the short-term prediction of traffic speed. An ANN model is also developed and a comparison of the performance of both these techniques is carried out, along with real time and historic approach results. Data from the freeways of San Antonio, Texas were used for the analysis.**

I. INTRODUCTION

INTELLIGENT Transportation Systems (ITS) is gaining more importance in the recent years as its level of deployment increases. The vast majority of urban transportation systems in North America are equipped with traffic surveillance systems that provide real time traffic information to traffic management centers (TMC), based on which the TMCs provide traffic information back to the travelers in real time. Such information is critical to both transportation system users and providers. The information provided to travelers through ATIS can be classified into three distinct groups: historical, real-time, and predictive. Historical, as its name implies, is based on archived data, while real time is based on the current values obtained from the system. Predictive is the predicted future values calculated using the real time or historic information. For pre-trip planning and en-route decisions, it is argued that predicted information would be more useful than real-time or historic information. The accuracy of this information is important since the travelers make appropriate decisions based on these information. Thus, short term prediction of traffic parameters including flow, speed and travel time is important. Prediction of speed and travel time has received considerable attention in the past few years as a means to support the Advanced Traveler Information systems (ATIS) and traffic management systems. Accurate forecasts of vehicle speed are also useful for evaluating the planning, design, operations and safety of roadways.

There are many research studies in the area of speed prediction [1]. In the context of prediction methodologies, different time series [2], [3] and ANN models [3], [4] have been developed. The objective of this study is to investigate the potential of Support Vector Machines (SVM) in predicting instantaneous traffic speed. It is also aimed at comparing and contrasting the performance of SVM with ANN. The predicted results are also compared with historic and real-time values.

II. DATA

The data were collected from the TransGuide Project area in San Antonio, Texas, USA [5]. A schematic diagram of the freeway system of San Antonio, showing the position of the test bed is given in Fig. 1. The I-35 North freeway was selected, which is equipped with dual loop detectors at 0.5-mile intervals. The data were reported in 20-second intervals and includes flow, occupancy and speed. The data were analyzed over a continuous 24-hour period for five consecutive days starting from August 4, 2003, Monday to August 8, 2003, Friday. Data from detector stations 159.500, 159.998, and 160.504 mile-posts were selected for the analysis. After extracting the specific detector data from the whole set, an extensive data reduction and quality control process was carried out to identify and correct any discrepancies in the data. The quality control and analysis included checking for missing data and threshold checking on the speed, volume and occupancy observations, individually as well as in combination. After the data reduction and quality control, the data were aggregated into

Manuscript received March 1, 2004

Lelitha Vanajakshi is a doctoral candidate in the Dept. of Civil Engg., Texas A&M University, College Station, Texas 77840, USA. Phone: 979 845 9881, fax: 979 845 648, e-mail: lelitha@yahoo.com

L. R. Rilett is with the Department of Civil Engg., Texas A&M University, College station, Texas 77840, USA (e-mail: rilett@tamu.edu).

2-minute intervals in order to reduce the level of data to be processed while still capturing most of the trends in the varying traffic conditions. Thus, an original file with data for a 24-hour time period having around 4300 records was reduced to 720 records after aggregation.

Sample results are shown from location 159.998 for illustration. The other detector locations also gave similar results. The data from all the five days at location 159.998 were compared and it was found that except Monday, all days had similar distribution as shown in Fig. 2. As can be seen, Monday data did not show any peak flow.

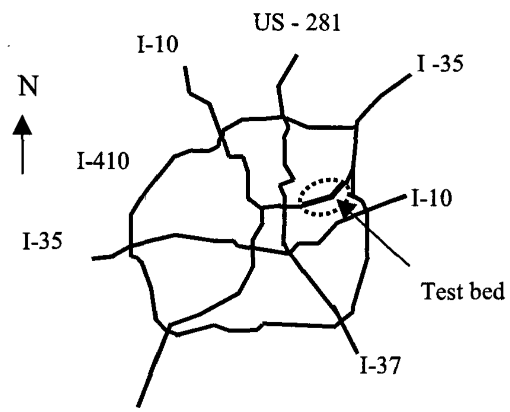

FIG. 1. Schematic diagram of San Antonio freeway system.

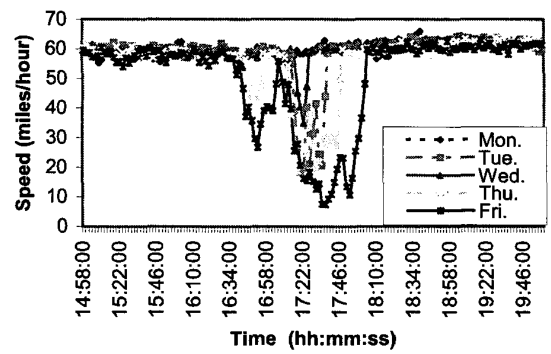

FIG. 2. Sample speed distribution on the study dates.

III. Models For Traffic Speed Prediction

The first ATMS simply used historic and/or real-time approach for forecasting speeds and travel times. Later, more sophisticated models including neural networks were used. Since ANN, historical, and real time methods are used for comparison purposes in this study, a brief discussion of each of these is given in the following subsections, in addition to the SVM method.

A. Historical and Real Time Method

The historic approach is based on the assumption that the historic speed profile can represent the traffic characteristics for a given time of the day. Thus, a historical average value will be used for predicting future values. An important component of the historic approach is the classification of days into day-types with similar profiles. The speed is defined by:

$$\overline{U}(t)=\sum_{i=1}^{n}U(i,t), \tag{1}$$

where, n is the number of days of data from past, $U(i,t)$ is the past speed at time t of day i.

This method can be valuable in the development of prediction models since they explain a substantial amount of the variation in traffic over time periods and days. However, for the same reason, the reliability of the prediction is limited because of its implicit assumption that the projection ratio remains constant [6].

Commuters in general have an idea about the average traffic condition and will be interested in conditions under abnormal conditions, that is, when average values are not representative of the current or future traffic conditions. In the real time approach, it is assumed that the speeds from the data available at the instant when prediction is performed represent the future condition. Thus, speed at one time step ahead is given by:

$$U(t,\Delta)=U(t), \tag{2}$$

and speed at two time step ahead is given by:

$$U(t,2\Delta)=U(t), \tag{3}$$

where, Δ is the time step and $U(t)$ is the speed at the present time t.

This method can perform reasonably well for the prediction into immediate near future under traffic flow conditions without much variations [7].

B. ANN

Artificial Neural Network (ANN) is a computing technique, which can be trained to learn a complex relationship in a data set. Basically it is a parallel computing system composed of interconnecting simple processing nodes [8], [9]. The ANN model, with its learning capabilities, is suitable for solving complex problems like prediction of traffic parameters. In particular, multi-layer feed forward neural networks that utilize a back propagation algorithm have been applied successfully for forecasting traffic speed and other traffic parameters [3], [4], [10].

Hence, in this study a multi-layer feed forward neural network with back propagation algorithm was used. Back propagation is a supervised learning algorithm that provides a method to adjust the weights in a multi layer network of connected processing units. The back propagation algorithm is an extension of the least mean square (LMS) algorithm, which will minimize the errors between the actual and the desired output.

Back propagation performs a gradient descend in the weight space to minimize the error at the outputs. This is done by calculating the error function for each input pattern and then back propagating the error from one layer to the previous one. The weights of a node are adjusted in direct proportion to the error in the units to which it is connected. Any of the measures of error such as the sum of the mean square error can be used for this purpose.

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x'_j, \tag{4}$$

where,

$w_{ij}(t)$ = weight of the connection between the node i and node j at time t,

x'_j = either output from node j or the input to the network,

η = gain term, and

δ = error term for node j.

In this study, the back propagation algorithm neural network was coded in MATLAB. One hidden layer with 10 neurons was found to be the optimum in the present study.

A major perceived disadvantage of ANN models is that, unlike other statistical models, they provide no information about the relative importance of the various parameters [10]. In ANN, as the knowledge acquired during training is stored in an implicit manner, it is very difficult to come up with reasonable interpretation of the overall structure of the network [8]. This lead to the term "black box" which many researchers use while referring to ANN's behavior.

C. SVM

At present ANN is one of the most popular methods in use for the prediction of traffic parameters. However, there are numerous practical shortcomings associated with conventional ANN including the difficulty in selecting the optimum number of hidden layers and hidden neurons. Another common concern about ANN is the difficulty in providing a reasonable interpretation of the overall design of the ANN network, as discussed already. In response, a number of modifications are proposed to alleviate these shortcomings [11]. These shortcomings led to explore alternative techniques such as SVM. SVM is a recently developed pattern classification and regression technique, which is being successfully applied to a number of applications ranging from particle identification to database marketing [12].

Since SVM is a relatively new technique, a brief explanation of how it works is given below using an example of a binary classification problem (Ch.2 of [8]). In this case SVM attempts to place a linear boundary between the two different classes, and orient it in such a way that the margin is maximized. For this the learning problem is cast as a constrained nonlinear optimization problem. In the case of classification of linearly separable data, the approach is to find among the hyper planes the one that minimize the training error as shown in Fig. 3. The SVM tries to orient the boundary such that the distance between the boundary and the nearest data point in each class is maximal as shown in Fig. 4. The boundary is then placed in the middle of this margin between the two points. Maximal margin is used for better classification of new data (generalization). The nearest data points are used to define the margins and are known as support vectors. Once the support vectors are selected, the rest of the data can be discarded [13]. Thus, SVM uses the strategy of keeping the error fixed and minimize the confidence interval.

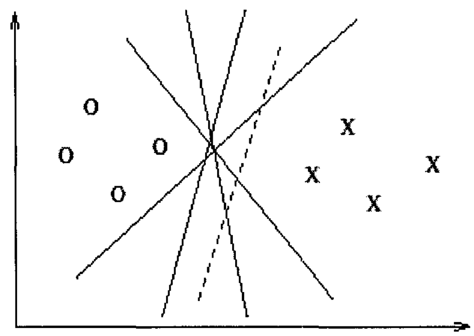

FIG. 3. Separating hyper planes

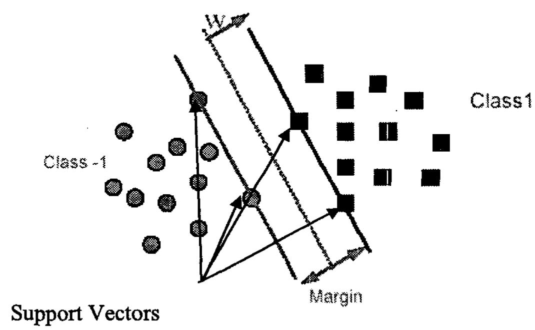

FIG. 4. Support vectors with maximum margin boundary

This can be explained mathematically as follows. Let the binary classification data points be

$$D = \left\{(x^1, y^1), \ldots (x^l, y^l)\right\}, \quad \mathbf{x} \in \Re^n, \mathbf{y} \in \{-1, 1\}. \tag{5}$$

where,

y = a binary value representing the two classes, and

x = the input vector.

As explained already, there are a number of hyper planes that can separate these two set of data and the problem is to find out the one with the largest margin. The SV classifiers are based on the class of hyper planes (boundary line),

$$(\mathbf{w}.\mathbf{x}) + b = 0, \quad \mathbf{w} \in \Re^n, b \in \Re, \tag{6}$$

where,

w = the boundary,
x = the input vector, and
b = the scalar threshold.

To remove redundancy, the hyper plane is considered in canonical form defined by a unique pair of values (w,b) at the margins satisfying the condition:

$$(\mathbf{w}.\mathbf{x}) + b = 1, \tag{7}$$

$$(\mathbf{w}.\mathbf{x}) + b = -1. \tag{8}$$

The quantities w and b will be scaled for this to be true, and therefore the support vectors correspond to the extremities of the data. Thus, the decision function that can be used to classify the data is:

$$\mathbf{y} = sign((\mathbf{w}.\mathbf{x}) + b). \tag{9}$$

Thus, a separating hyper plane in canonical form must satisfy the following constraints:

$$y_i\left[(\mathbf{w}.\mathbf{x}_i) + b\right] \geq 1, \quad i = 1,...l. \tag{10}$$

where,
l = the number of training sets.

There can be many possible hyper planes that can separate the training data into the two classes. However, the optimal separating hyper plane is the unique one that not only separates the data without error but also maximizes the margin. This means that it should maximize the distance between the closest vectors in both classes to the hyper plane. This margin, ρ is the sum of the absolute distance between the hyper plane and the closest training data points in each class.

This distance *d (w,b;x)* of a point *x* from the hyper plane *(w,b)* is:

$$d(w,b;x) = \frac{\left|(\mathbf{w}.\mathbf{x}_i) + b\right|}{\|\mathbf{w}\|}. \tag{11}$$

Thus, the sum of the absolute distance between the hyper plane and the closest training data points in each class *i* and *j*, ρ is calculated as given in equation 14.

$$\rho = \min \frac{\left|(\mathbf{w}.\mathbf{x}_i) + b\right|}{\|\mathbf{w}\|} + \min \frac{\left|(\mathbf{w}.\mathbf{x}_j) + b\right|}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}. \tag{12}$$

The optimal canonical hyper plane is the one that maximizes the above margin. Thus, the optimal hyper plane, with the maximal margin of separation between the two classes can be uniquely constructed by solving a constrained quadratic optimization whose solution is in terms of a subset of training patterns that lie on the margin. These training patterns, called support vectors, carry all relevant information about the classification problem.

Thus, SVMs are constructed from a unique learning algorithm that extracts training vectors that lie closest to the class boundary, and makes use of them to construct a decision boundary that optimally separates the different classes of data. In the simplest form, SVM use a linear separating hyper plane to create a classifier with a maximal margin. In cases where the given classes cannot be linearly separated in the original input space, the SVM first non-linearly transforms the original input space into a higher dimensional feature space. This transformation is carried out by using various non-linear mappings: polynomial, sigmoidal etc. After the non-linear transformation step, SVM finds a linear optimal separating hyper plane in this feature space [8], [12]. In Support Vector regression (SVR) the basic idea is to map the data into a high-dimensional feature space via a non-linear mapping and to do linear regression in this space. Thus, linear regression in a high dimensional (feature) space corresponds to non-linear regression in the low dimensional input space.

In the present study, Support Vector Regression was selected for the prediction of speed. The SVR model used a radial basis kernel function. The value of insensitivity function ε was selected to be 0.05. This value was selected by trial and error. SVM toolbox for MATLAB developed by Gunn was used for the present study [14].

IV. Results

First, the 2-minute aggregated data were normalized based on the range of the speed values on each day. The input and output data were selected as the speed for the 5 previous time step values and the speed for the next time step value respectively. Because the data were grouped in 2-minute intervals, five time steps corresponds to a 10-minute interval. Thus, the prediction was based on the previous 10-minute speed values. The model then estimated the next 2-minute speed. The prediction was subsequently carried out to 4 minutes, 6 minutes, *etc.* up to an hour ahead. The training data were varied from one full day data to four days data and testing was done for a separate day. Results obtained from location 159.998 are shown here for illustration. The other detector locations also gave similar results.

FIG. 5. MAPE for prediction using 1-day data for training

Fig. 5 shows the error in prediction when a single day data (Monday) were used for training the network and Friday data were predicted. Fig. 6-8 show similar result when the training data was increased to 2-day, 3-day and 4-day (Monday to Thursday) and the Friday data were predicted. The performance based on the historical, real time, ANN and SVM methods are shown in these figures.

The performance measure used was Mean Absolute Percentage Error (MAPE) as shown in (13).

$$\text{MAPE} = \frac{\sum \frac{|actual - estimated|}{actual}}{Number\ of\ observations} \times 100. \tag{13}$$

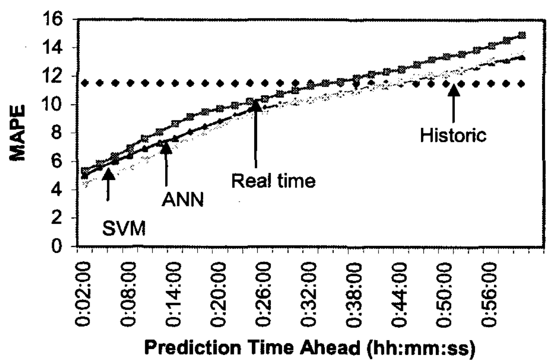

FIG. 6. MAPE for prediction using 2-day data for training

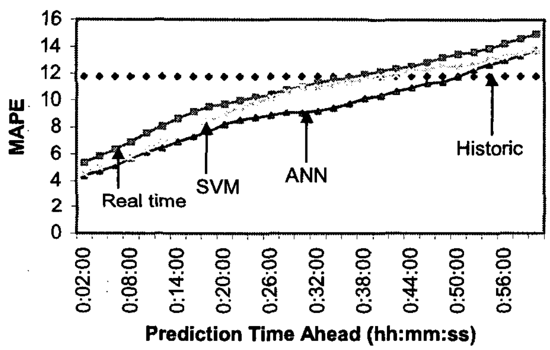

FIG. 7. MAPE for prediction using 3-day data for training

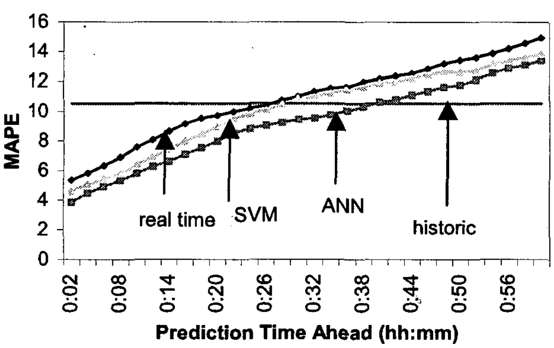

FIG. 8. MAPE for prediction using 4-day data for training

It can be seen that the SVR method performed better than ANN when only a single day's data were used for training. However, when the training data were increased to two, three and four days, the performance of ANN gradually improved and became better than SVM as shown in Fig. 6, 7, and 8. This may be due to the difference in pattern of Monday data compared to other days. In the case of Monday data alone for training, the training data and testing data were not having similar pattern and the amount of training data also is less. Under such situations SVM performed better than ANN. This can be explained based on the inherent nature of the ANN and SVM training process. SVM method is independent of the training data, once it chooses the data points that can represent the input data (support vectors). Hence, if one day training data contains enough support vectors, the reduction in error due to more training data may not be significant. However, in the case of ANN, the network can learn more about the data as the training data increase and this leads to better results. This can be seen further in Fig. 9 which shows the variation in MAPE with respect to the training data used, namely training with 1-day data, training with 2-days data *etc.* for a 4-minute ahead prediction using ANN and SVM techniques. It can be seen that the change in error with more training data is not very significant in the case of SVM.

The actual speed values and the corresponding predicted values for a 2-hour evening peak and off-peak period for a 2-minute ahead prediction is shown in Fig. 10 for illustration. It can be seen that ANN and SVM was able to follow the trends in the actual data under varying conditions. It can also be seen that the historic method failed when the speed data had more variations such as under peak flow conditions. Real time method showed a constant shift from the actual values through out.

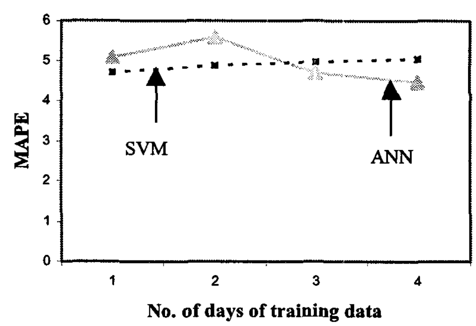

FIG. 9. Variation in MAPE with respect to the training data used for a 4-minute ahead prediction.

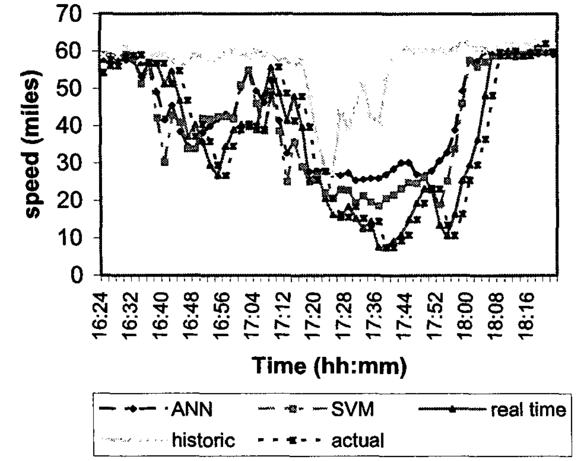

FIG. 10. Predicted speed values for a 2-minute ahead prediction with 2 days data for training.

V. Summary and Conclusions

This paper presented a comparison of performance of ANN and SVM for the short-term prediction of traffic speed. The ANN model used was a multi-layer feed forward neural network and the SVM model used was a support vector regression with radial basis kernel function. The analysis considered forecasts ranging from 2 minutes ahead up to an hour into the future. One full day data were left for cross validation to evaluate the prediction errors. The training data were varied from one single day data to four days data. The results were compared with historic and real time approach results.

Results of this comparison indicate that the explanatory power of Support Vector Regression is comparable to ANN. Also, SVR performed better than ANN when the training data were less in quality and quantity. While more investigation is necessary to draw definite conclusions regarding the advantages of SVM to ANN, based on the investigations conducted in this study, it was found that SVR is a viable alternative to ANN for short-term prediction. The performance of ANN depends largely on the amount of data available for training the network. Thus, in situations where the available data are less, and the training data is not a good representation of the whole data, there is a need for an alternative method for prediction. Due to the characteristic nature of the SVM method, the performance of SVM is almost independent of the training data, once the network chooses the support vectors. Hence, when the training data are non-representative of test data, SVM outperformed ANN. In other cases, the performance of SVM was comparable to ANN, thus making it an alternative option for prediction problems.

The study also showed that current traffic conditions are good predictors while long-range predictions need the use of historical data. The ANN and SVM methods performed well for some range into future. Also, both these methods have good dynamic response and show better performance compared to the traditional models. The training of both SVM and ANN may not make them attractive for online applications. However, both of them can be trained offline, and then used for on-line prediction. Once the networks are trained and the network parameters are stored off-line, the system can be used for online-applications, where the travel time corresponding to the incoming data need to be predicted quickly.

To the knowledge of the authors there have been very few studies that explored the use of SVM in transportation applications and there have been none, which used SVM for the prediction of traffic variables. Lot more work is needed to exploit the explanatory power of this powerful tool to the fullest. Also, more research concerning the effect of each of the different parameters of SVM such as kernel function and cost function on the prediction performance is needed.

References

[1] S. Lee, D. Kim, J. Kim, and B. Cho, "Comparison of models for predicting short-term travel speeds," *5th World Congress on Intelligent Transp. Systems*, Korea, 1998.

[2] D. Mahalel, and A. S. Hakkert, "Time series for vehicle speeds", *Transportation Research*, Vol. 19B, No. 3, 1995, pp. 217 – 225.

[3] J. Mc Fadden, W. T. Yang, and S. R. Durrans, "Application of artificial neural networks to predict speeds on two-lane rural highways," *Presented at the 80th TRB Annual Meeting*, 2001.

[4] S. H. Huang, and B. Ran, "An application of neural network on traffic speed prediction under adverse weather condition," *Presented at the 82nd TRB Annual Meeting*, 2003.

[5] *Transguide Model Deployment Initiative Design Report and transguide Technical Paper*, Texas Department of Transportation, <http://www.transguide.dot.state.tx.us/ PublicInfo/papers.php>, (November 1, 2003).

[6] C. Hoffman, and J. Janko, "Travel time as a basis of the LISB guidance strategy," *In Proceedings Of IEEE Road Traffic Control Conference*, IEEE, New York, 1990, pp. 6-10.

[7] P. Thakuriah, A. Sen, J. Li, N. Liu, F. S. Koppelman, and C. Bhat, "Data Needs For Short Term Link Travel Time Prediction," *Advance working paper series number 19*, Urban Transportation Center, University of Illinois, Chicago, 1992.

[8] V. Kecman, *Learning And Soft Computing: Support Vector Machines, Neural Networks, And Fuzzy Logic Models*, The MIT press, Cambridge, Massachusetts, London, England, 2001.

[9] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 1999.

[10] D. Park and L. R. Rilett, "Forecasting freeway link ravel times with a multi-layer feed forward neural network," *Computer Aided Civil And Infra Structure Engineering*, vol. 14, 1999, pp. 358 – 367.

[11] M. S. Dougherty, H. R. Kirby, and R. D. Boyle, "Using neural networks to recognize, predict and model traffic," *Artificial Intelligence Applications to Traffic Engrg.*, M. Bielli, G. Ambrosino and M. Boero, Eds., Netherlands, 1994, pp. 233 – 250.

[12] C. Campbell, "Kernel Methods: A Survey Of Current Techniques," *Neuro Computing*, Vol. 48, 2002, pp. 63-84.

[13] B. Samanta, "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms," *Mechanical Systems and Signal Processing*, Vol.18, 2004, pp. 625 - 644.

[14] S. R. Gunn, "Support Vector Machines for Classification and Regression," http://www.ecs.soton.ac.uk/~srg/ publications/ pdf/SVM.pdf, <Nov. 23, 2003>